1-1-2007

Curve reconstruction based on the relative neighbourhood graph.

Augustus Kumar Das
University of Windsor

Curve Reconstruction
Based On
The Relative Neighbourhood Graph

by

Augustus Kumar Das

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the School of Computer Science
in Partial Fulfillment of the Requirements for the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada
2007

Your file Votre référence
ISBN: 978-0-494-42306-6
Our file Notre référence
ISBN: 978-0-494-42306-6

Abstract

This thesis work deals with the problem of curve reconstruction in a novel way. Here the relative neighbourhood graph on a set of points is used and a heuristic for curve reconstruction is proposed. Given a sample point set S, first a Relative neighbourhood graph is constructed on S, which is proved to contain all edges joining adjacent points along the unknown curve. Next, the non-adjacent edges are removed to find the reconstruction. Experimental results for sample points drawn from a variety of curves is provided. A second heuristic dealing with curve reconstruction in presence of noise is proposed. Noise points are unwanted points that creep in during data collection. The second heuristic takes a noisy sample as input and produces a filtered set of points to which the first heuristic can be applied to find the reconstruction.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. Asish Mukhopadhyay, without whose support and guidance this thesis work would have been impossible to accomplish. His continuous support and insightful guidance helped in every stage of my master's degree, I feel privileged to have him as my advisor. I would like to sincerely thank Dr. Dan Wu for his valuable suggestions to improve my work. I also extend my thanks to Dr. Myron Hlynka for spending his valuable time on my thesis evaluation committee and proofreading my thesis.

I gratefully acknowledge my colleagues Samidh Chatterjee and Demin Yin for their suggestions and feedback which helped me in many ways.

Dedicated to my mother Hemalata Das and my sister Adrika..

Contents

List of Figures

Chapter 1

Introduction

The problem of reconstructing a curve, surface or a shape in general is a central one in many problems and has applications in science and engineering. In its simplest form, the problem can be stated as follows:

Given a set of points which are taken from an unknown shape as a sample, the problem is to recreate the unknown shape by connecting the sample points that describe the shape.

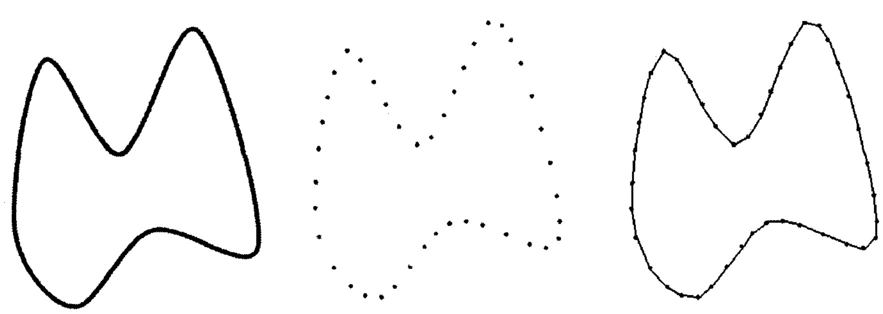


Figure 1.1: *The original Curve, Sample Points and the reconstructed Curve*

Researchers have come up with algorithms that provide guarantees on the quality of reconstruction provided the input satisfies some sampling conditions.

1.1 Multi-Disciplinary field

The problem of reconstructing an unknown shape from sample data has been studied extensively in a variety of disciplines among which are:

- Computational Geometry.
- Computer Graphics.
- Geometric Modeling.
- Image Processing.
- Computer Vision.

1.2 Classification

The general problem of shape reconstruction can be classified according to the dimensionality of the problem. The most important applications arise in two and three dimensions and accordingly it can be classified as:

- Two-Dimensional Curve Reconstruction.
- Three-Dimensional Surface Reconstruction.

Algorithms addressing the shape reconstruction problem in higher dimensions are mostly theoretical in nature. However most researchers tend to generalize their algorithms to dimensions higher than three.

1.3 Problem Statement

The curve reconstruction problem is to reconstruct an unknown curve C in a given class (for example smooth and closed, or smooth and open etc.) from a sample S of n points $\{p_1, p_2, p_3, \ldots, p_n\}$ (See Fig. 1.2).

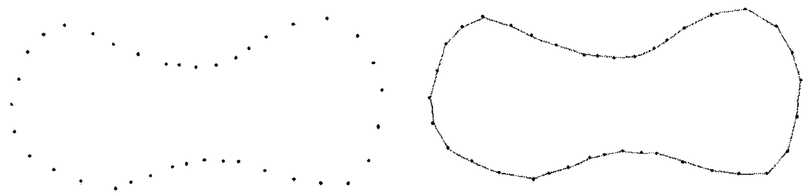


Figure 1.2: *Sample Points (S) and the reconstructed Curve(C)*

1.4 Application areas

The curve reconstruction problem finds applications in areas like:

- Pattern recognition.
- Computer vision
- Cluster analysis

More importantly, solutions to the curve reconstruction problem serve as a stepping stone for solutions to the more complicated Surface reconstruction problem, which in turn finds applications in diverse areas of science and engineering. Some examples are:

- Bio-Medical research and instruction.
- Solid Modeling.
- Geology.
- Archeology.
- Geographical Information System.
- Oceanography.
- Industrial Inspection.
- Reverse engineering.

1.5 Terminology

1.5.1 Voronoi diagram

The Voronoi diagram of n points in two dimensions is a partition of the plane into convex regions such that each region contains exactly one generating point and any point in this region is closer to its generating point than to any of the other point in the plane (See Fig. 1.3). Aurenhammer [6] discusses the importance and utility of Voronoi diagrams.

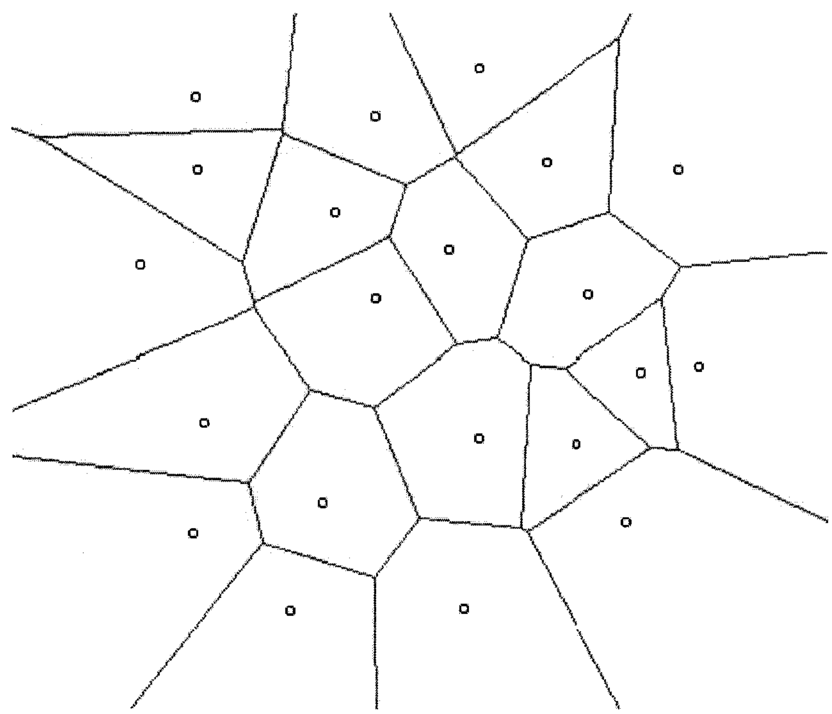


Figure 1.3: *Voronoi Diagram on a set of points*

1.5.2 Delaunay Triangulations

The Delaunay graph is the face-dual of the Voronoi diagram. When no four points are co-circular, this graph is a triangulation of the point set, called the Delaunay triangulation. The circumcircle of each triangle in this triangulation has no other points in its interior (See Fig. 1.4).

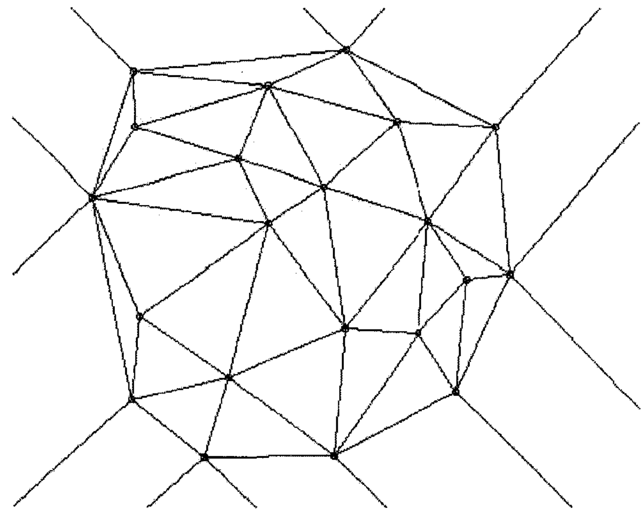


Figure 1.4: *Delaunay Triangulation of a set of points*

1.5.3 Medial Axis

The medial axis of a smooth curve is the locus of the centers of maximal disks that touch the curve at two or more points (See Fig. 1.5). Every point on the medial axis has two or more closest points on the curve.

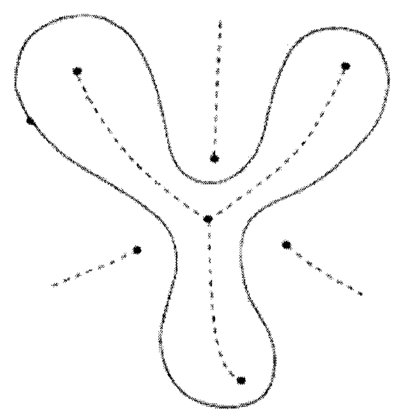


Figure 1.5: *Medial axis of a curve [15]*

1.5.4 Local feature size

The local feature size, $f(x)$ at a point x on the curve C is the distance from x to the nearest point on the medial axis (See Fig. 1.6).

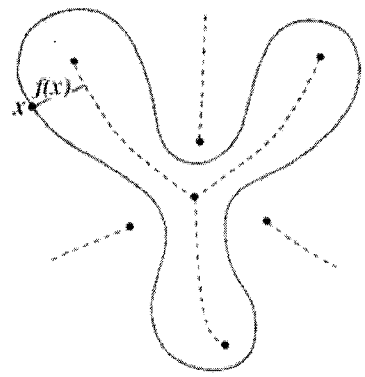


Figure 1.6: *Local feature size f(x) at x [15]*

1.5.5 Sampling conditions

Sampling conditions constitute a restriction or a set of restrictions on the density and quality of sampling of input points. In order to guarantee faithful reconstruction this is essential.

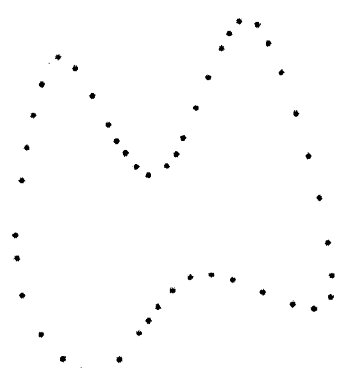


Figure 1.7: *Sample points of Curve in fig.1.1 showing variable sampling density*

A set of sample points drawn from a curve is said to be an ϵ-sample if, for any point p on the curve there is a sample point within a distance of ϵ times LFS(Local feature size) at p. For correct reconstruction it is necessary that ϵ <1. Some algorithms for curve reconstruction require the sample points to be uniformly sampled whereas others allow non-uniform sampling. The intuition behind this is that, we can have fewer sample points at parts of the curve with less detail and more sample points at parts with more details, intersections, corners etc.(See Fig. 1.7).

1.5.6 Relative neighbourhood graph (RNG)

The *Relative Neighbourhood Graph* (RNG) was introduced into the Computational Geometry literature by Toussaint [36]. For a given point set S, an edge joining two points p and q is in the RNG if the lune defined by the edge does not contain any other point from the sample set S(see Fig. 1.8).

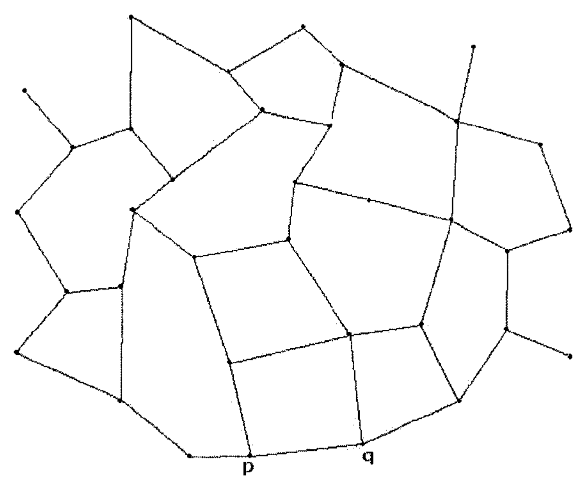


Figure 1.8: *RNG on a point set*

The lune is the region of intersection of the circles centered at p and q, each with radius equal to length of the edge $\overline{pq}$ (see Fig. 1.9).

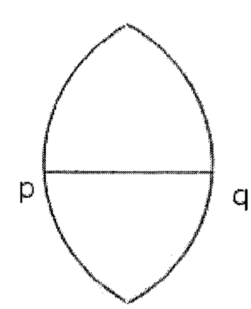


Figure 1.9: *A Lune*

1.5.7 Gabriel graph

The *Gabriel graph* was introduced into the Computational Geometry literature by Gabriel and Sokal [26]. The Gabriel graph of a set of sample points S consists of the set of edges joining pairs of points, such that the diametral circle on each such edge contains no other points of S(see Fig. 1.10 and Fig. 1.11). This is also a sub-graph of the Delaunay triangulation on S and a supergraph of the RNG. For more details about neighbourhood graphs see Jaromczyk and Toussaint [25].

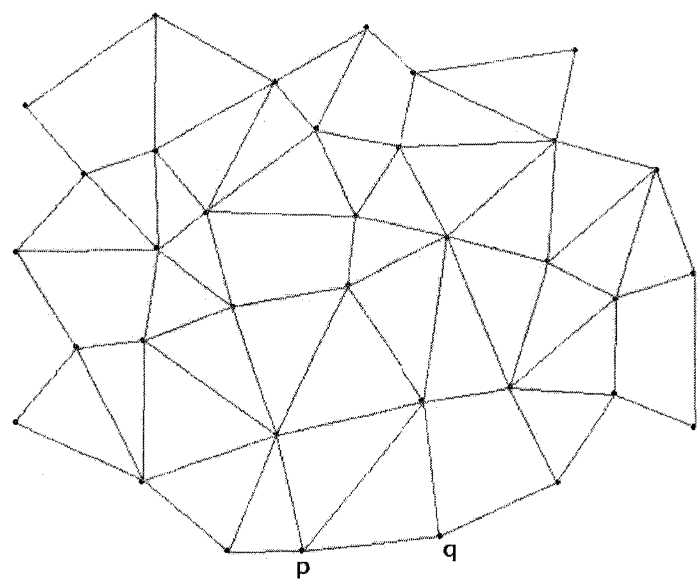


Figure 1.10: *Gabriel graph on a point Set*

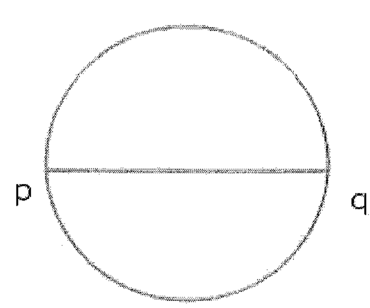
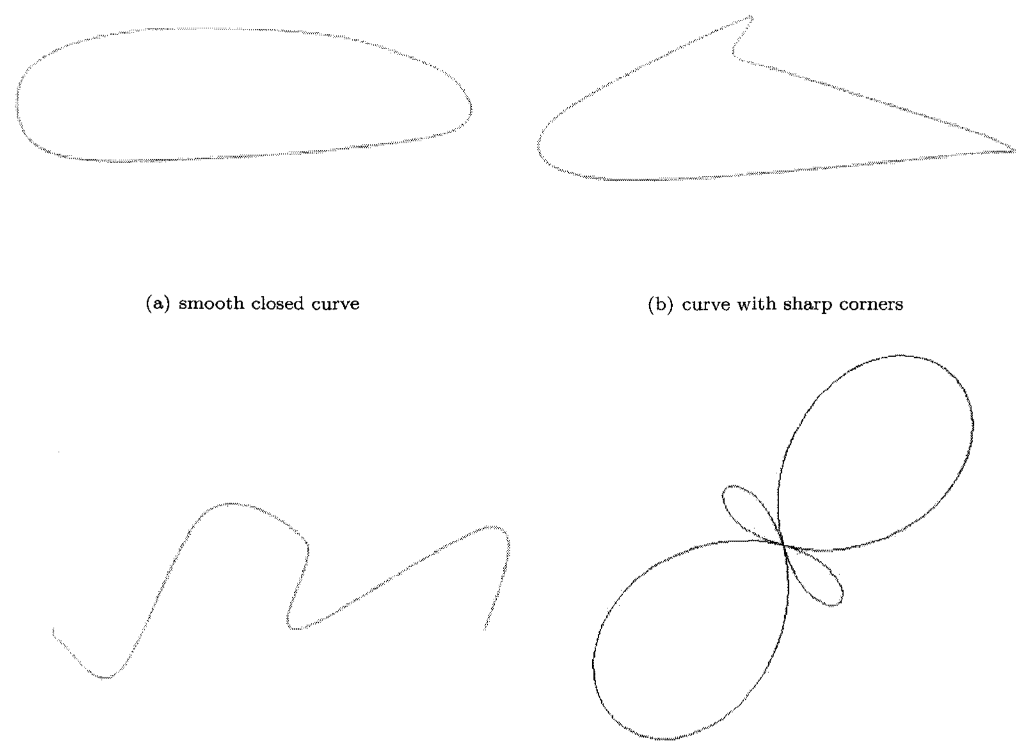


Figure 1.11: *Diametral circle*

1.5.8 Types of curves

The complexity and efficiency of the reconstruction algorithm depends on the type of curve it deals with. Some classes of curves are (See Fig. 1.12):

- Smooth closed curve.
- Curve with sharp corners.
- Curve with Endpoints.
- Curve with Intersections.



(a) smooth closed curve (b) curve with sharp corners

(c) curve with end points (d) curves with intersections

Figure 1.12: *Types of Curves*

1.6 Motivation

The thrust of our work on the problem of curve reconstruction is three-fold, and can be stated as follows:

- To find a new subgraph of the Delaunay triangulation which retains all the edges of the correct reconstruction.
- To start with a subgraph that has fewer edges to be removed to arrive at the correct reconstruction.
- An algorithm that is easy to understand and implement.

1.7 Contribution

Our contribution to this area of computer science can be summarized as follows:

- Proving that the RNG of the set of sample points retains all the edges adjacent on the reconstructed curve.
- A new heuristic for reconstructing curves from sample points.
- A new heuristic for filtering noise from a noisy sample.

Chapter 2

Literature Review

The curve-reconstruction problem has received lot of attention from researchers over the last decade. In this section, I will discuss some of the work by researchers that have helped evolve the problem of curve reconstruction, and has been instrumental in solving the more difficult problem of surface reconstruction in three dimensions. Most of the algorithms proposed to solve the problem can be broadly classified as:

- Delaunay based approaches.
- Non-Delaunay based approaches.

2.1 Delaunay Based approaches

Many researchers have made use of one of the most significant data structures in computational geometry called the Delaunay triangulation as well as its dual - the Voronoi diagram. Some significant algorithms proposed in the literature that handle the curve reconstruction problem are as follows:

- Alpha Shapes.
- r-regular shapes.
- Crust.
- Nearest Neighbor.
- Conservative crust.
- Traveling Salesman approach.

2.1.1 Algorithms based on α-shapes

One of the earliest attempts to classify the shape of a set of points in space was made by Edelsbrunner et al.[17] in 1983. He generalized the convex hull of a finite set of points to lead to a family of straight line graphs called α-shapes. He proposed the idea of α-shapes and α-hulls and showed that α- shapes captured the 'fine shape' and 'crude shape' of a set of points in a plane.

Their path-breaking paper gives a generalized and comprehensive analysis of the convex hull of a finite set. He defined an entire family of straight-line graphs called α-shapes which capture the heuristic notion of 'fine shape' and 'crude shape' of point sets. It is shown in their paper that α-shapes are subgraphs of closest or furthest point Delaunay triangulation. Based on this result they develop an optimal $O(n \log n)$ algorithm for constructing α-shapes. At the end they point out the flexibility of the α-shapes in being generalized to 3 or more dimensions. They also address the question of dynamization which states that given the α-shapes of set S for some α, how does the insertion of a point into S or deletion of a point from S affect the α-shape.

2.1.2 Algorithms Based on r-regular shapes

Many of the ideas surrounding some of the recent Delaunay triangulation-based reconstruction algorithm originate in the work by Brandt and Algazi [9], who showed how to obtain the skeleton of an r-regular shape from the Voronoi diagram of a set of points sampled along the boundary of that shape.

The parameter r controls two aspects of such a shape: the curvature at a boundary point never exceeds the reciprocal of r, and the radius of a maximal disk contained in this shape or its complement never exceeds r. Moreover, they were able to show that for shapes in this class the computed skeleton converges to the exact skeleton as the sampling density increases.

Dominique Attali [5] borrowed this notion of an r-regular shape and provided a correct reconstruction for shapes in this class under some guarantee on the sample. Her ideas were further refined and amplified by subsequent researchers.

2.1.3 Crust Algorithm

Amenta, Bern and Epstein [2] showed how to reconstruct a smooth curve from a non-uniform ϵ-sample. In this important paper, the authors describe two graphs called the crust and the β-skeleton for reconstruction of smooth curves. The condition that they put on the sampling is that the distance between two consecutive points be at most r times the local feature size, where $r < 1$. When this sampling condition is met, the two graphs produce a correct polygonal reconstruction of a curve.

They present a two-step algorithm to find a graph, which they called the crust. The intuition behind their idea of the crust was that the Voronoi diagram on a set of sample points approximate the medial axis of the curve and the Voronoi disks of $S' = S \cup V$ where S is the set of sample points and V is the set of Voronoi points obtained from constructing the Voronoi diagram on S roughly represent the empty circles between a curve and its medial axis.

They define an edge belonging to the Delaunay triangulation of S' as belonging to the crust if both its endpoints belong to the sample point set S. In this case they proved that when $\epsilon \leq .253$, the crust correctly reconstructs the unknown curve(See Fig. 2.1).

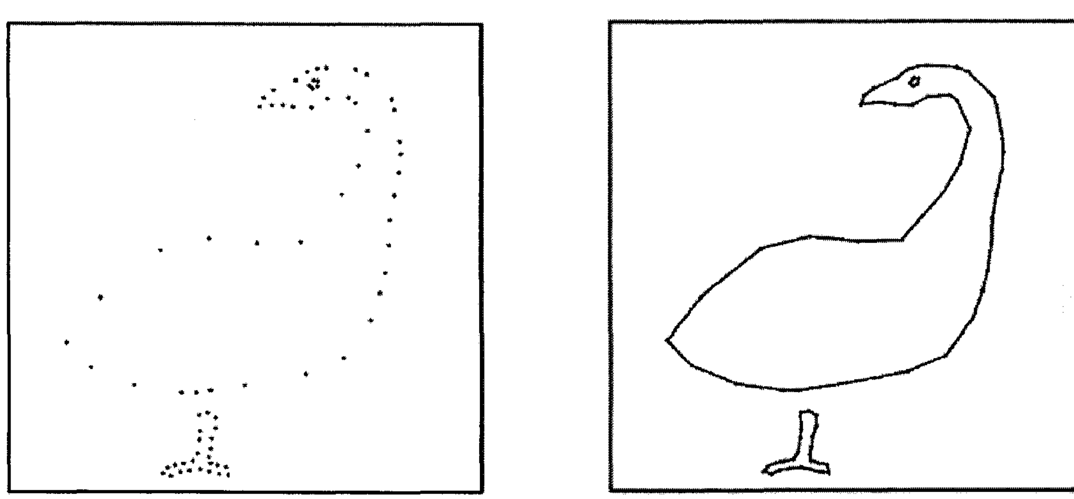


Figure 2.1: *Curve Reconstruction by Crust Algorithm [2]*

For β-skeleton they use the concept of forbidden region and the circle based β-skeleton. If s_1 and s_2 are two points then the union of two disks of radius $\beta *$ distance$(s_1, s_2)/2$ touching both points defines the forbidden region for the two points, where $\beta > 1$. An edge joining s_1 and s_2 belongs to the β-skeleton if and only if the forbidden region is empty. They guarantee reconstruction for $\beta = 1.70$.

Their most important contribution was the idea of local feature size $f(p)$ at a point p on the unknown curve $\mathcal{C}$, which determines the sampling density in the neighbourhood of p.

2.1.4 NN-crust Algorithm

Dey and Kumar[12] proposed a very simple algorithm for curve reconstruction using the nearest neighbour graph which they call the NN-crust algorithm. The nearest neighbour graph is formed by connecting the edges that connect sample points in the point set to its nearest point. They prove that the edges connecting a point to its nearest neighbour are present in the reconstruction. However this technique does not find all the edges. They complete the reconstruction by adding what they call half-neighbour edges.

They find the half-neighbour edges by finding a point that is incident on only one edge e of the NN graph. These are breaks where edges are to be added. They find the shortest edge incident on this point among all edges of the Delaunay triangulation that make an angle of more than $\pi/2$ with e. The union of the nearest neighbour edges and these new edges gives the complete reconstruction.

This algorithm is very simple and it improves on the sampling density. They guarantee correct reconstruction for $\epsilon \leq 1/3$.

2.1.5 One Step Algorithm

Gold [22] presents a one step algorithm to find two graphs called the crust and anti-crust, which he uses to solve issues related to map input analysis. The crust is equivalent to the polygonal reconstruction of the curve and the anti-crust is equivalent to the medial axis of the curve.

He makes use of quad-edge data structure for his solution. His approach is local rather than global. He considers each quad edge and apply the crust test to it. If it passes the crust test he assigns it to the crust, else he retains it as part of the skeleton or anti-crust, which is equivalent to the medial axis of the curve. He avoids the use of a second structure that was needed to find the crust by [2]. In the process he finds both graphs wanted in one step. However the locally defined crust found sometimes contains additional edges other than the globally defined edges as found by Amenta et al. [2].

2.1.6 Conservative Crust Algorithm

In another significant paper Dey et al. [13] propose a different algorithm for the polygon reconstruction of a curve from a sample. They expanded the work to include curves that are not necessarily smooth and closed. They go on construct a smooth curve from the polygonal reconstruction obtained in the previous step in a step called curve fitting.

The work of this thesis is closely related to this work. They use the Gabriel graph as a starting point to find a polygonal reconstruction. Then they refine the Gabriel graph by eliminating edges from the Gabriel graph to form a new graph G'. An edge e is retained in the new graph if a ball B centered at the midpoint of the edge with radius equal to $length(e)/\rho$ is a Voronoi Ball empty of Voronoi vertices.

They further refine the graph by eliminating an edge e for which $X = G' \cap B$, where B is a ball with center at midpoint of e and radius equal to $length(e)/4 * \rho$, contains a degree 0 vertex or a degree 1 vertex not connected to the edge e within X. They output the final graph as the polygonal reconstruction when no such edge remains. They introduce the additional parameter ρ by which they control the quality of reconstruction.

In another paper Dey and Wenger [14] propose a heuristic for reconstructing curves with are not smooth and have multiple components. They claimed the algorithm handles sharp corners better than the existing algorithms.

Most algorithms control reconstruction by imposing restrictions on the sample points. For curves with corners it is practically impossible to get such a sample as the sampling needs to be infinitely dense at corner points. So they defined a different sampling condition for sharp corners.

2.1.7 Other Algorithms

From this point on, the focus shifted to build algorithms that can handle a wide variety of curves and guarantee reconstruction for curves which are more complex in nature like curves with sharp corners and endpoints, a collection of curves etc.

Funke and Ramos [19] proposed an algorithm to reconstruct, not just one curve, but a collection of curves under certain sampling conditions. They also include curves with sharp corners and endpoints in their collection of curves. They too define two different sampling conditions for smooth sections and corner sections of the curve. They start by identifying edges that they call smooth and then explore corner sections and if needed change the status of some of the edges earlier classified to be smooth.

While all of the above papers we have referred to are based on the Delaunay triangulation, Guha and Tran [23] proposed a non-Delaunay-based technique that reconstructs a curve. We have not explored non-Delaunay based approaches in this thesis and our focus in this thesis is on Delaunay-based approaches for curve reconstruction.

Chapter 3

Curve Reconstruction with RNG

This chapter describes the main contribution of this thesis. The intuition was that a different subgraph of the Delaunay triangulation can serve as a better starting point to find the polygonal reconstruction. The Relative Neighbourhood Graph(RNG) was chosen for this purpose.

Fig. 3.1 shows the RNG as well as the Gabriel graph on the same set of sample points. As evident from the picture, the Gabriel graph contains many more edges joining non-adjacent points than the RNG. So starting from the RNG we have fewer redundant edges to be removed.

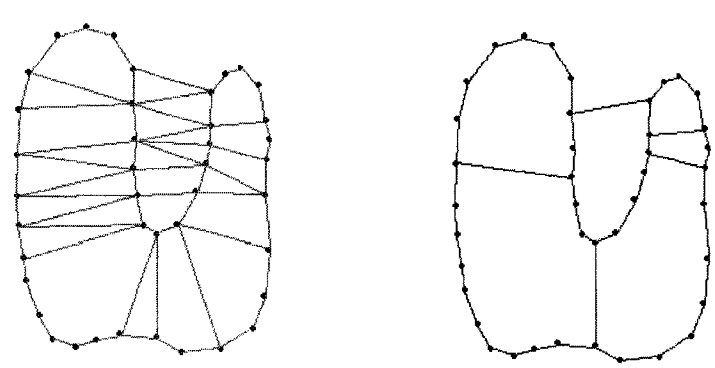


Figure 3.1: *A Gabriel graph and an RNG on the same set of sample points*

In the next section we prove that RNG does retain all the adjacent edges along the target curve. In the following section a new heuristic for curve reconstruction is presented. The algorithm is supported by a curve reconstruction applet that can be found at:

`http://cs.uwindsor.ca/~asishm/publicDomainSoftware`
`/RNGcurveReconstruction/RNG4max.html`

3.1 RNG captures the good edges

Here we show that we can start with the RNG instead of the Gabriel graph and still go on to find the correct reconstruction. The algorithm proposed is closely related to the work of [13]. They extract the Gabriel graph from the Delaunay triangulation of the sample point set S, and show that it contains all the edges that connect adjacent points along the curve in the final reconstruction, provided S is an ϵ-sample with $\epsilon < 1/3$.

To establish that the RNG contains all the adjacent edges needed for reconstruction we need the results of some lemmas from [2] and [12]. First the statement of these lemmas and their proofs follow. The lemmas were proved by the respective authors and have been restated here for continuity of reading.

Lemma 3.1. *Let S be an ϵ-sample from a smooth, planar curve C and H be its polygonal reconstruction from S. Then*

1. *If a closed disk B contains 2 or more points of C, then its intersection with B is either a topological 1-disk or contains a point of the medial axis [2].*
2. *For $\epsilon < 1$, if ab is an edge of H, then $length(ab) < 2\epsilon/(1-\epsilon) * f(p)$, where $f(p)$ is the local feature size at p and p is an endpoint of the edge (can be a or b).[12].*

1. **Proof of 1** [2]: Let us assume $B \cap$ C is not a topological 1-disk, because if $B \cap C$ happens to be a topological 1-disk, then we are done. Otherwise if some connected component of $B \cap C$ is a closed loop in the interior of B, then it contains a point of the medial axis and the lemma is proved. Fig. 3.2(a) shows some types of intersection of curve F with disk B. (The figure illustrates the

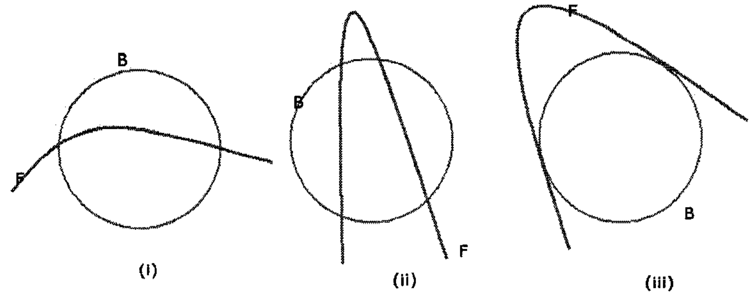
cases when the intersection is a topological 1-disk (i), and two other cases (ii) and (iii) where the intersections have a medial axis point inside B). The case of a closed loop (Jordan curve) in the interior of B is not shown.

If $B \cap C$ has two or more connected components, let us consider o to be the midpoint of B (See Fig. 3.2(b)). Suppose p is the closest point on F to o . Now if we have more such points on F, which are closest to o, then we have two or more points on F closest to o and o is a point on the medial axis. If p is the only closet point to o on F then, let q be the closest point to o on another connected component. Now, if we take a point x on the line segment between o and q, then the point x is closer to o than any point outside the disk B. Hence the closest point of F to x is always on a connected component of F, rather than outside B. Now we have o which is the closest to one connected component and the other connected component has a closest point on the line segment between o and q. So if we move along the segment, at some point x the closet connected components change. At this point x, we have two closest points to F and hence we have a point of the medial axis contained inside B and the lemma is proved.

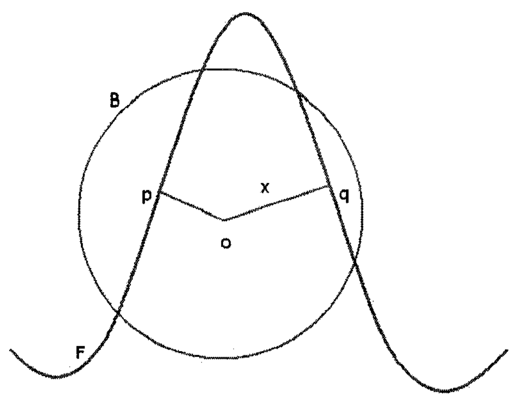
2. **Proof of 2** [12]: We have an edge ab and a point q where the perpendicular bisector of this edge meets the curve over which a and b are adjacent. See Fig. 3.2(c). Now, keeping its center at q we grow a ball (stages of growth shown as (i), (ii) and (iii)), until it meets the two endpoints a and b of the edge ab. Now this ball always intersects the curve as a topological 1-disk, failing which the ball would have a radius greater than or equal to the feature size at q, $f(q)$, the moment it touches the first endpoint. This follows from the previous lemma because if it was not a topological 1-disk then it would have two connected components and a point of the medial axis. But this contradicts the ϵ sampling condition with $\epsilon < 1$. So, it indicates that the two endpoints a and b are the nearest samples to q. This gives $length(ab) < 2\epsilon * f(q)$. Now for any two points p and q on a smooth curve C we have from (Lemma 3.1) in [12]:

$$f(q) \leq f(p) + length(pq).$$

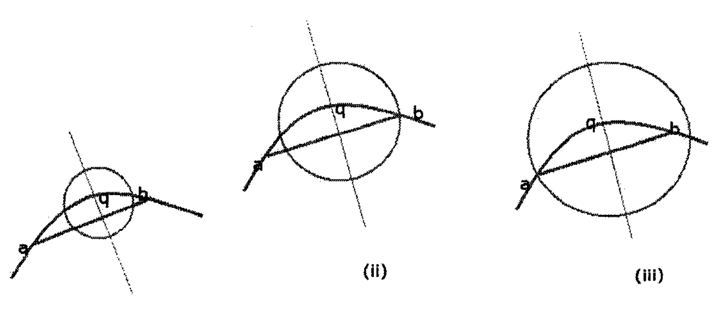
This can be derived using the triangle inequality. Further we can replace length(pq)



(a) Types of Intersection of a curve with a Disc



(b) Intersection of F with B



(c) Growing a Disc to meet end points

Figure 3.2: *Illustration of Lemma 3.1*

by $\epsilon f(q)$ in the inequality to get

$$f(q) \leq f(p) + length(pq) \leq f(p) + \epsilon f(q).$$

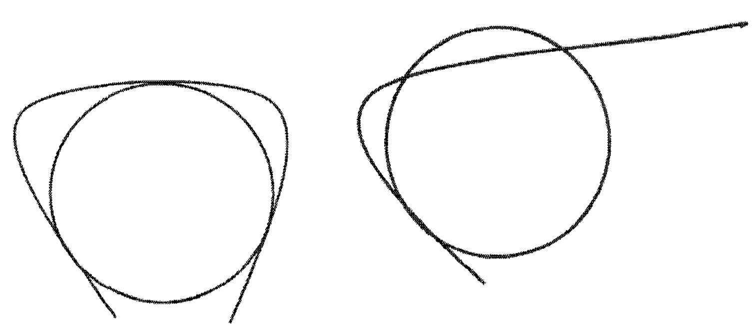
This gives us $f(q)$ which is less than $1/(1-\epsilon) * f(p)$ and hence

$$length(ab) < 2\epsilon/(1-\epsilon) * f(p).$$

Hence the lemma is proved.

Lemma 3.2. *Any Voronoi disk (a maximal empty disk centered at a Voronoi vertex of a Voronoi diagram) of a set of sample points S of a smooth, planar curve, $\mathcal{C}$, must contain a point of the medial axis of $\mathcal{C}$. [2]. This lemma does not hold in three dimensions.*

Proof: Let B is a Voronoi disk of S. Let s be one of the sample points on the boundary of B. Let C-s be completely contained in B in the neighbourhood of this sample s. Then there are two possibilites. Either $B \cap C$ is entirely contained in B and the center of the disk is a point of the medial axis or we can reduce B around its center to get a smaller disk B' entirely contained in B and $B' \cap C$ comprising two connected components and hence a point of the medial axis. If there are no s in S for which the above discussion holds, then we already have two connected components in B, thereby introducing a point of the medial axis inside it. Fig. 3.3(a)represents the first case where we have such a point s on the boundary and Fig. 3.3(b) represents the second where we already have two connected components.



(a) (b)

Figure 3.3: *Illustration of Lemma 3.2*

3.1.1 A new theorem.

Theorem 3.1. *Given an ϵ-sample S from a smooth closed curve $\mathcal{C}$, where $\epsilon < 1/5$, the graph H on S, obtained by joining successive points on the unknown curve C is contained in the RNG of S.*

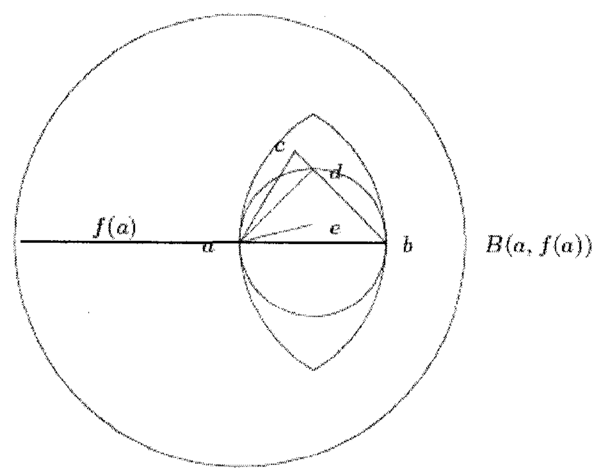


Figure 3.4: *An edge joining adjacent points along the curve is in the RNG*

Proof: Let ab be an edge joining two points a and b of S that are adjacent along the curve $\mathcal{C}$ (Fig. 3.4). If a point c of S witnesses the edge ab by being inside the lune defined by the edge ab, we consider two cases that can arise.

- c is inside the diametral circle of ab.

  We prove the impossibility of this case by using the same argument as in [13]. We briefly reproduce the argument here. Since $\epsilon < 1/5$ it is, a fortiori, less than 1 and hence $f(a) > length(ab)*(1-\epsilon)/2\epsilon$ by Lemma 1, part(ii). This means that the disk $B(a, f(a))$ contains the disk with ab as diameter. If c were inside this disk then it would have to intersect $\mathcal{C}$ in more than one component, and hence would have to contain a medial axis point by Lemma 1, part(i). This would imply that $B(a, f(a))$ would have to contain a medial axis point, contradicting the definition of $f()$.

- c is outside the diametral circle of ab.

  The triangle $\triangle abc$ is acute-angled since $\angle adb = 90$ degrees $> \angle acb$ (Fig. 3.4), while ac and bc being inside the lune, they cannot make an angle > 90 degrees with ab. Thus the center, e, of the circumcircle of $\triangle abc$ lies inside the triangle in question. Therefore, $2 * length(ae)$, the diameter of the circumcircle of $\triangle abc$ is $< 2 * length(ab)$. The last inequality is true because the point e is inside the lune of ab and hence inside the circle whose center is at a and has radius $=$ $length(ab)$.

  But then again by Lemma 1, part (ii), $f(a) > (1-\epsilon)/2\epsilon * length(ab) > (1-\epsilon)/2\epsilon * length(ae)$. We can therefore ensure that the circumcircle of $\triangle abc$ is inside $B(a, f(a))$ by letting $(1-\epsilon)/2\epsilon > 2$ or $\epsilon < 1/5$. We can now repeat the argument for the circumcircle of $\triangle abc$ as we did for the diametral disk of ab for the first case if it contains some other sample point; or if it is point-free it must be a Voronoi disk and hence contain a point of the medial axis by Lemma 2.

3.2 An RNG-based algorithm

From this point we proceed quite differently from the work done in paper [13]. What we have proposed is a simple heuristic for removing the edges that join pairs of non-adjacent points along the unknown curve C that works remarkably well. This section explains the working of the algorithm followed by a formal description of the RNG based curve reconstruction algorithm, which we call as the CR algorithm in short for curve reconstruction .

Fig. 3.5 shows a set of sample points derived from an unknown curve. Then we construct the RNG on the Set of sample points as seen in Fig. 3.6. Fig. 3.7 shows the Voronoi diagram constructed on the sample point set. Fig. 3.8 shows the Voronoi diagram superimposed on the RNG of the sample points.

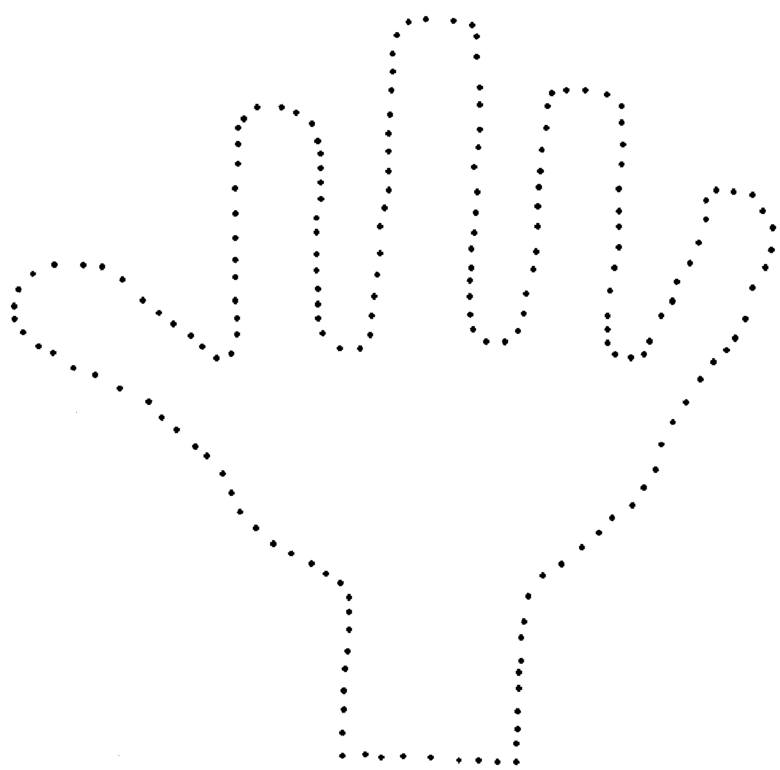


Figure 3.5: *Sample Points*

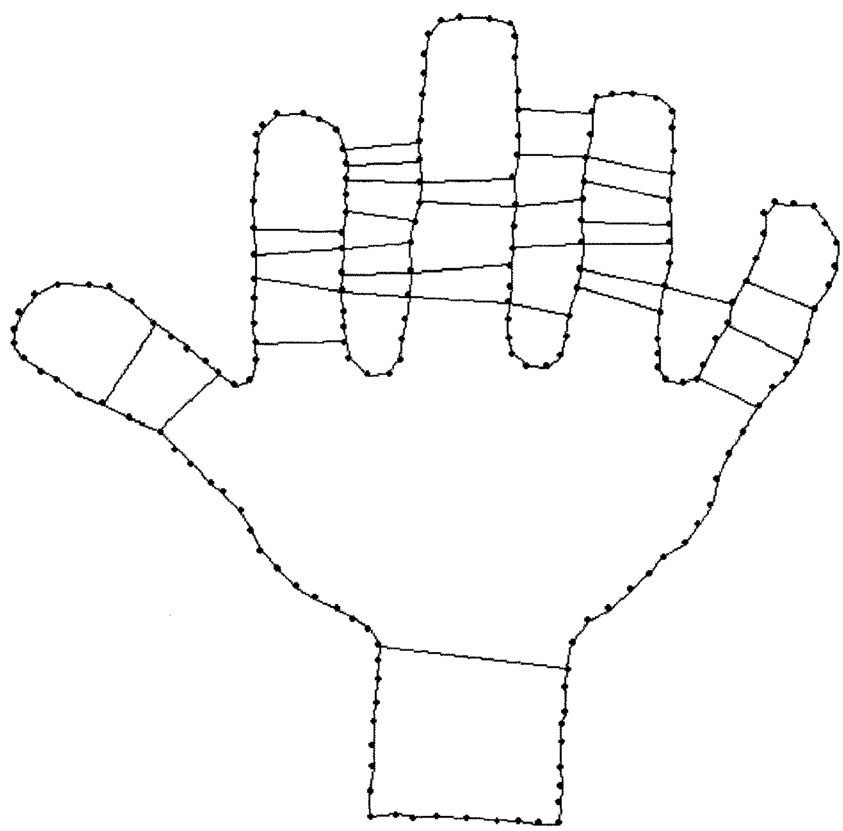


Figure 3.6: *RNG on the sample points*

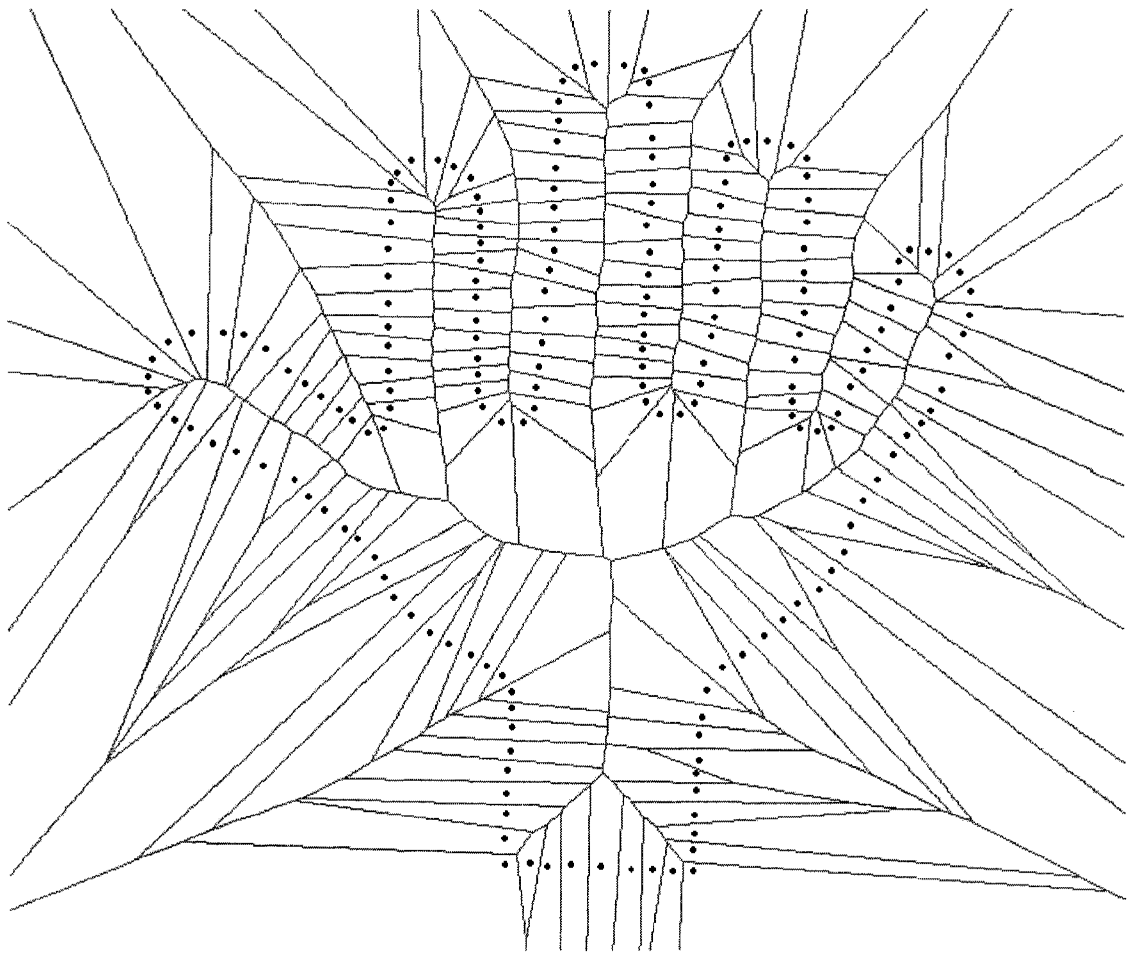


Figure 3.7: *Voronoi diagram of the sample points*

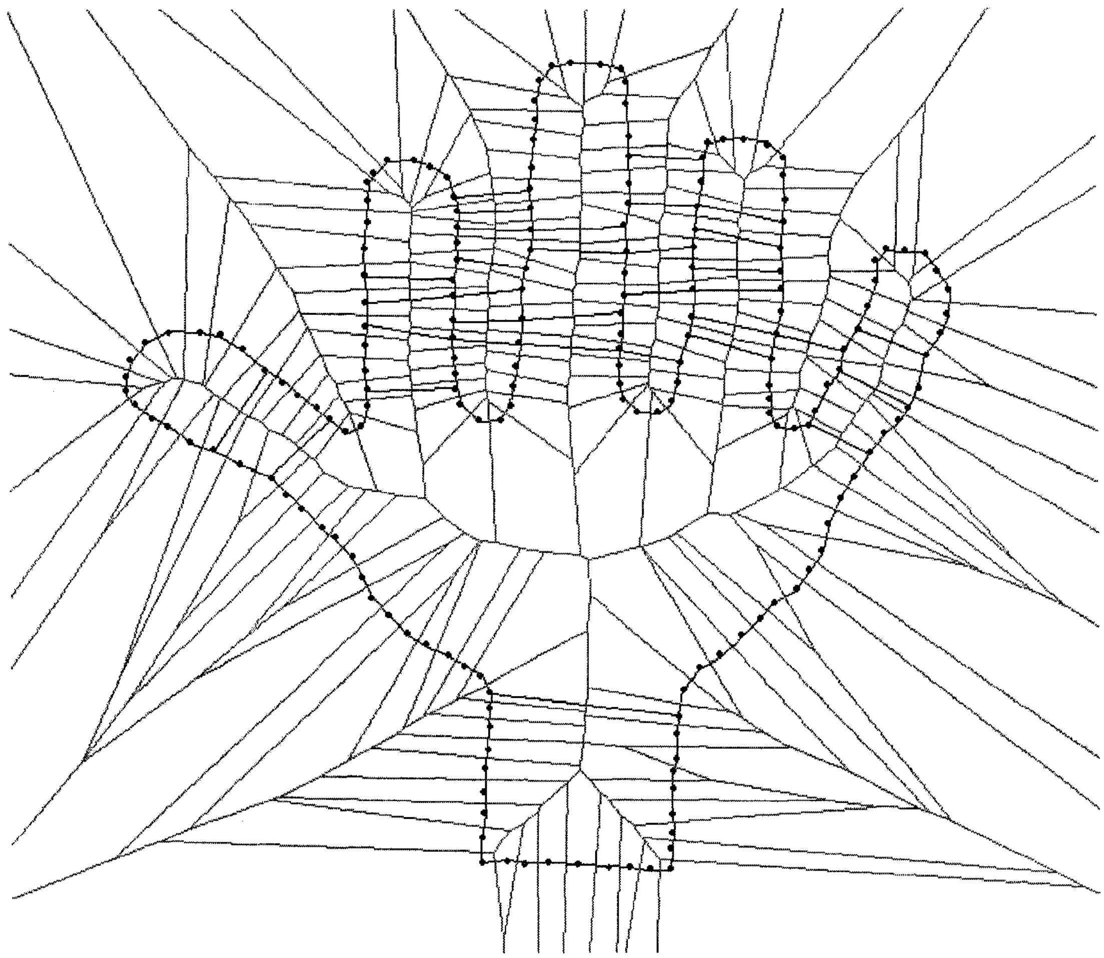


Figure 3.8: *RNG superimposed on the Voronoi diagram of the sample points*

Now, from the Voronoi diagram on S, for each RNG edge p_ip_j, we compute the maximum distance from p_i to the vertices of its Voronoi polygon. Fig. 3.9 shows a redundant RNG edge (for a different curve), the endpoints and the Voronoi polygon. Let this distance be d_i; do the same for the point p_j, obtaining a distance d_j. We delete the edge p_ip_j if its length is greater than the maximum of d_i and d_j. The rationale for this was based on experimentation with a large variety of samples. It turned out that the redundant edges always crossed the approximation of the medial axis due to the Voronoi diagram of a sample. The above heuristic attempts to estimate the distance from an end-point to the medial axis by computing the maximum distance from this point to a Voronoi vertex of its Voronoi polygon.

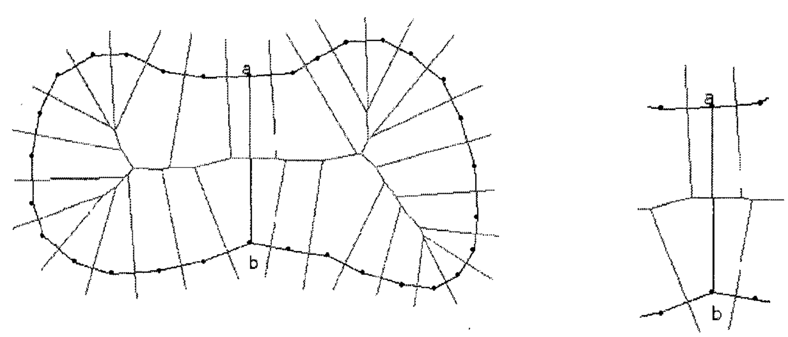


Figure 3.9: *Idea underpinning removal of an edge joining non-adjacent points*

When we have eliminated all redundant edges we have the polygonal reconstruction of the unknown curve (see Fig. 3.10).

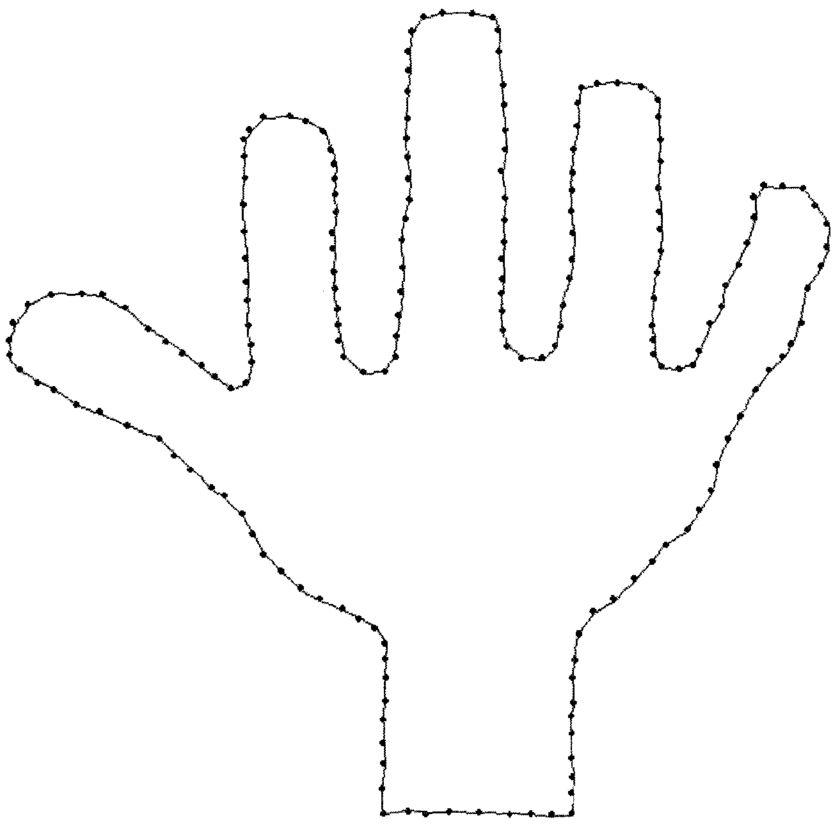


Figure 3.10: *Reconstructed curve*

Here is a formal description of the **CR Algorithm**.

Algorithm CR

Input: A set of sample points S from an unknown smooth curve C

Output: A polygonal reconstruction of C

Step 1. Compute the Delaunay Triangulation, DT on S.

Step 2. Extract the RNG from the DT.

Step 3. Compute the Voronoi Diagram, VD, as the dual of the DT obtained in Step 1.

Step 4. For each RNG-edge $\overline{p_i p_j}$ computed in Step 2 do:

Step 4.1 Compute the maximum distance d_i from p_i to the vertices of its Voronoi polygon.

Step 4.2 Compute the maximum distance d_j from p_j to the vertices of its Voronoi polygon.

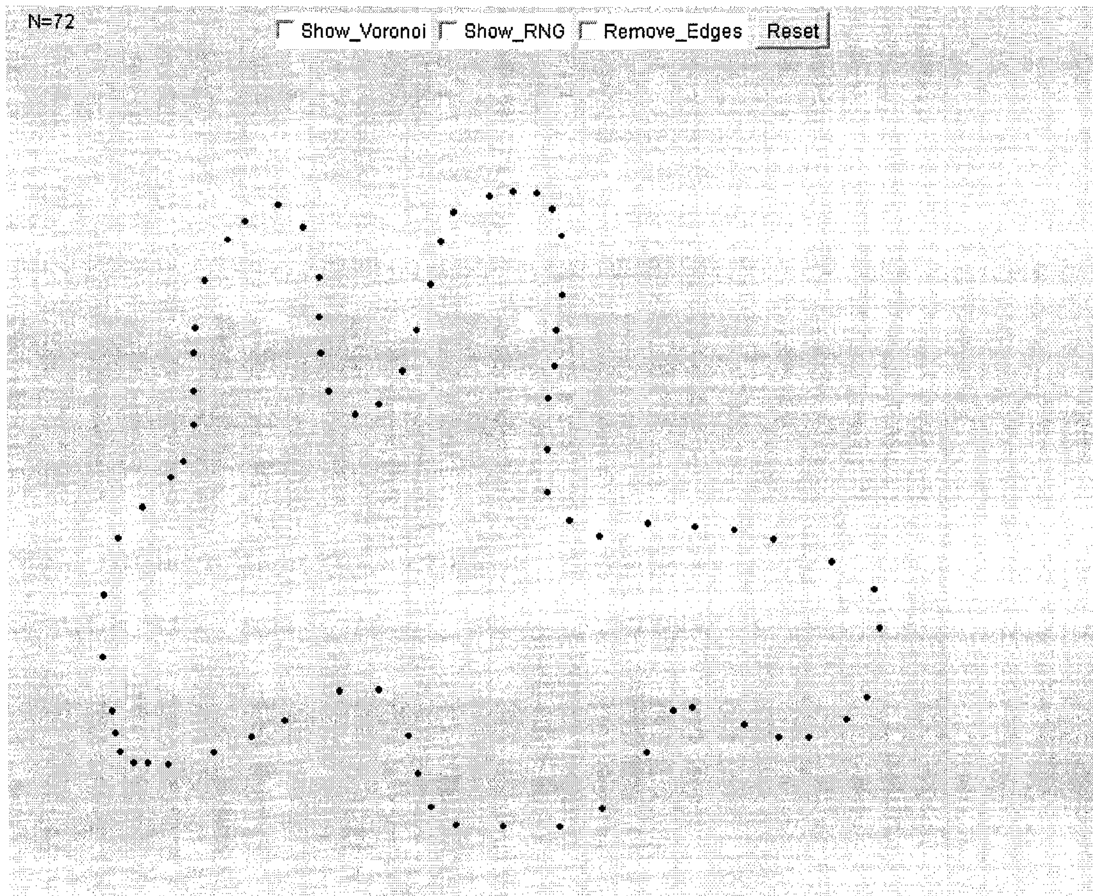
Step 4.3 Set $d_{max} = max(d_i, d_j)$.

Step 4.4 If $d_{max} < length(p_i p_j)$, delete edge $p_i p_j$.

Step 5. Output the remaining set of edges.

3.3 Implementation Details

The Algorithm was implemented with Java. The following pages show screen-shots from the implementation applet.



The Applet Reconstructs a Curve, or collection of curves (No intersections) from Sample Points Entered.

1. Enter Points that can be part of a curve, Use a Reasonably dense sample - Maximum 300 Points.
2. Draw_Voronoi draws the voronoi diagram on the set of points Entered
3. Draw_RNG draws the Relative Neighbourhood Graph (RNG) on the set of points Entered
4. Remove_Edges Removes the Extra Edges from the RNG and Shows the Reconstructed Curve
5. Use Reset to clear the present sample and Enter a Different Sample

Figure 3.11: *Sample points entered by the user*

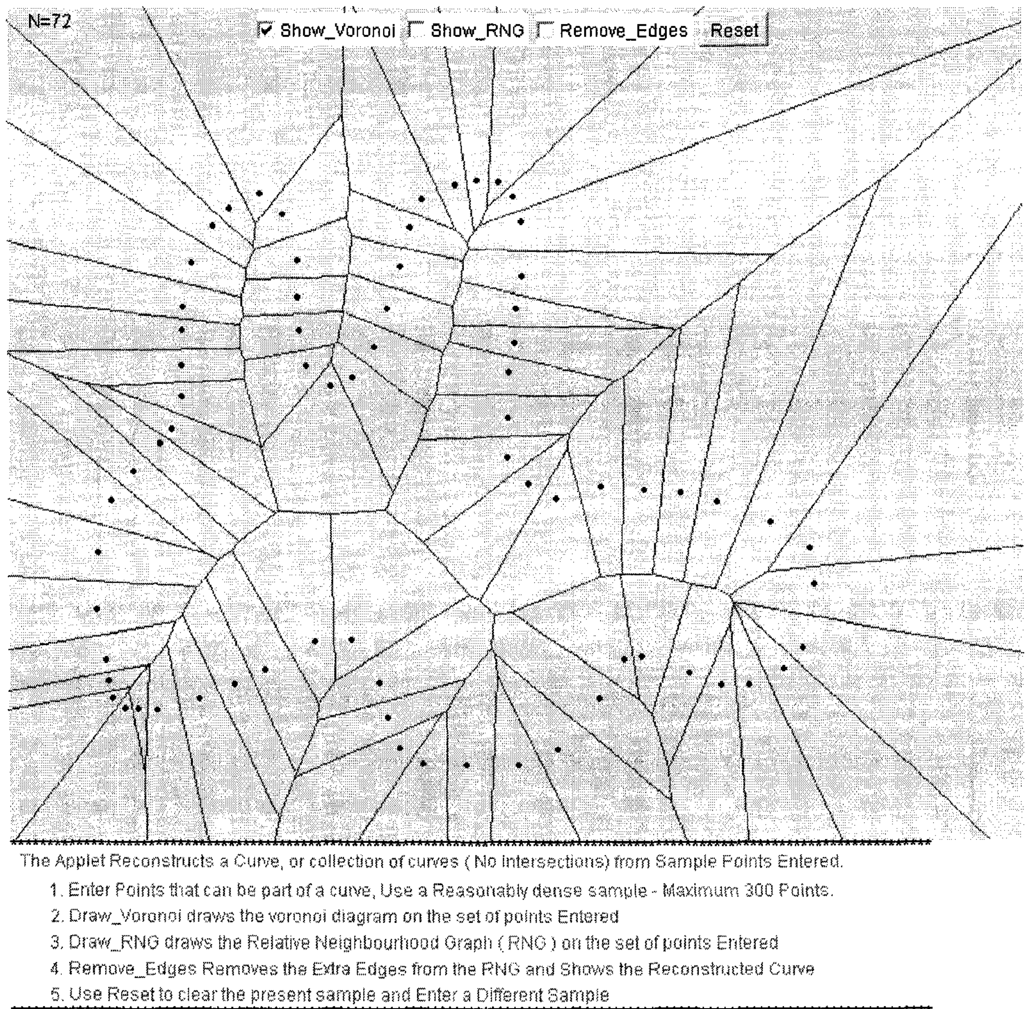


Figure 3.12: *Voronoi Diagram on the sample point set*

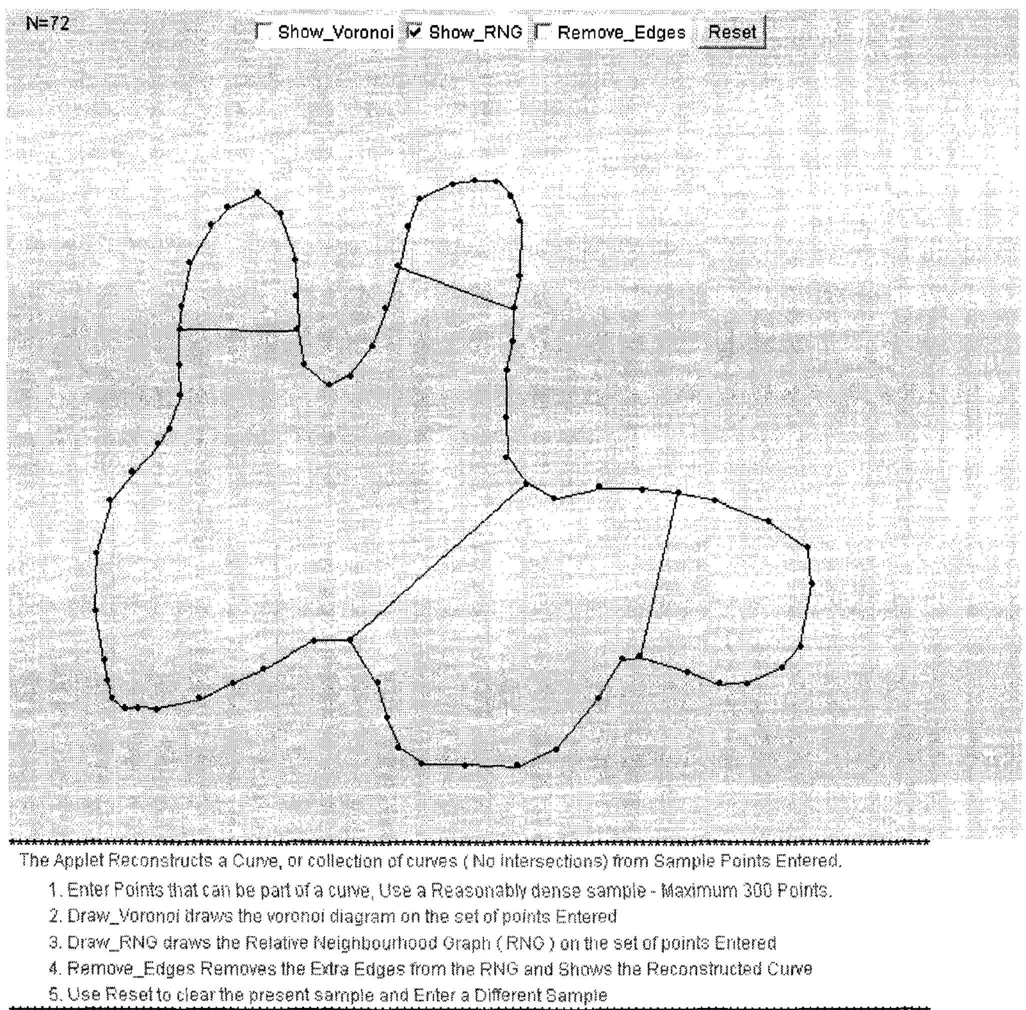


Figure 3.13: *RNG on the sample point set*

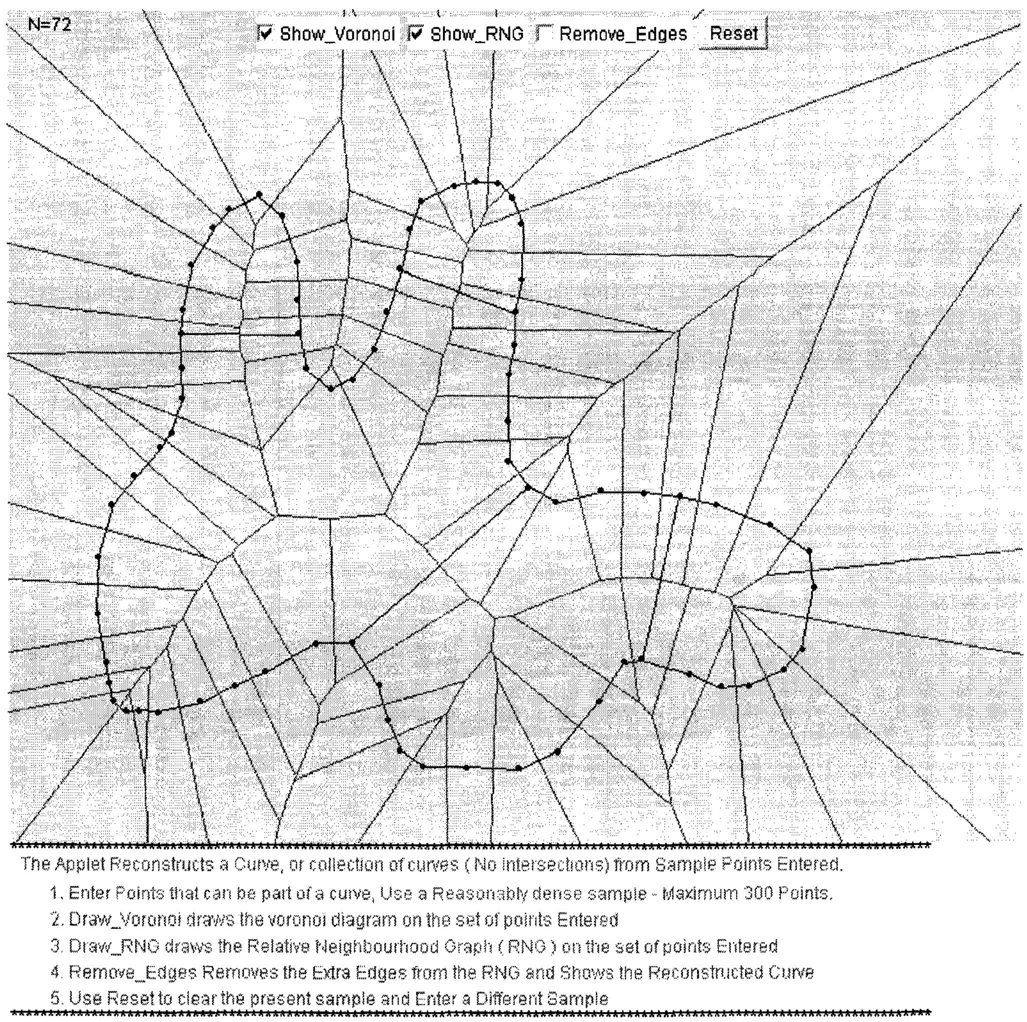


Figure 3.14: *RNG on the Sample Point set with the Voronoi diagram superimposed*

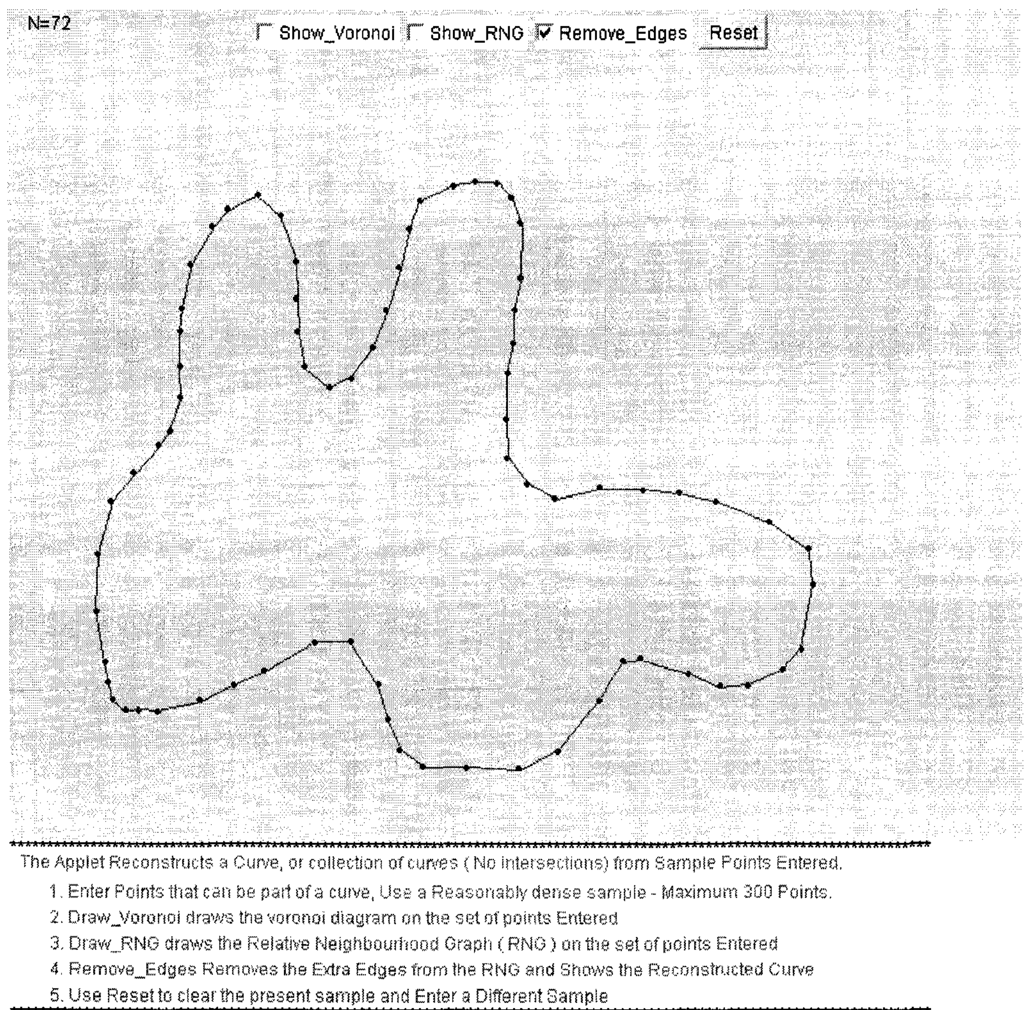
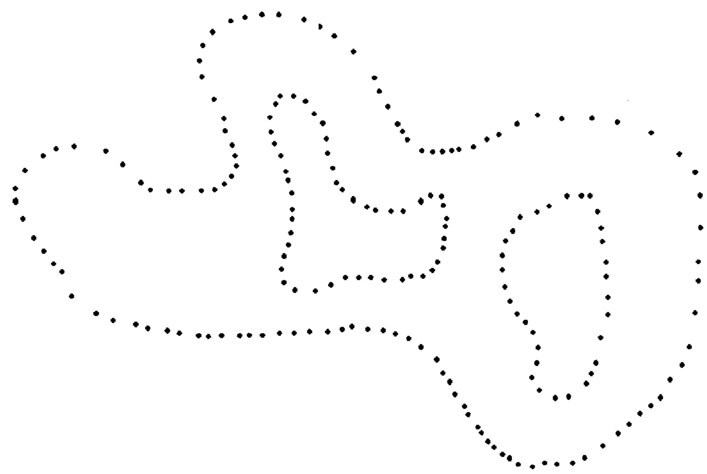


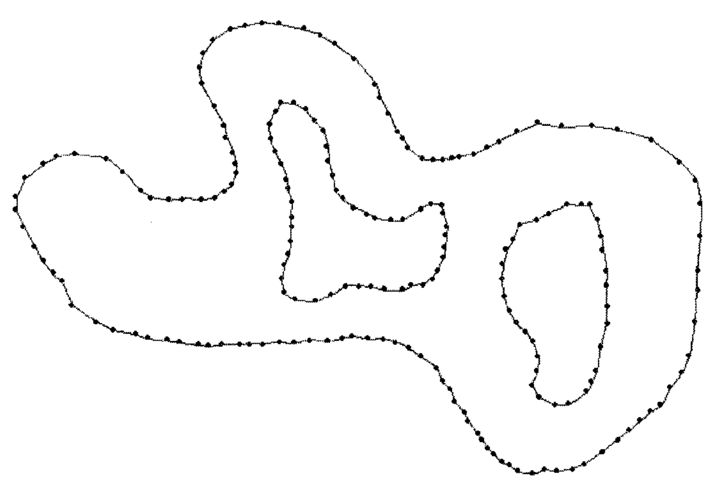
Figure 3.15: *Polygonal reconstruction of the unknown curve represented by the sample point set*

3.4 Examples of Reconstruction

This section shows results of implementation of the RNG Algorithm on sample points drawn from a variety of curves.

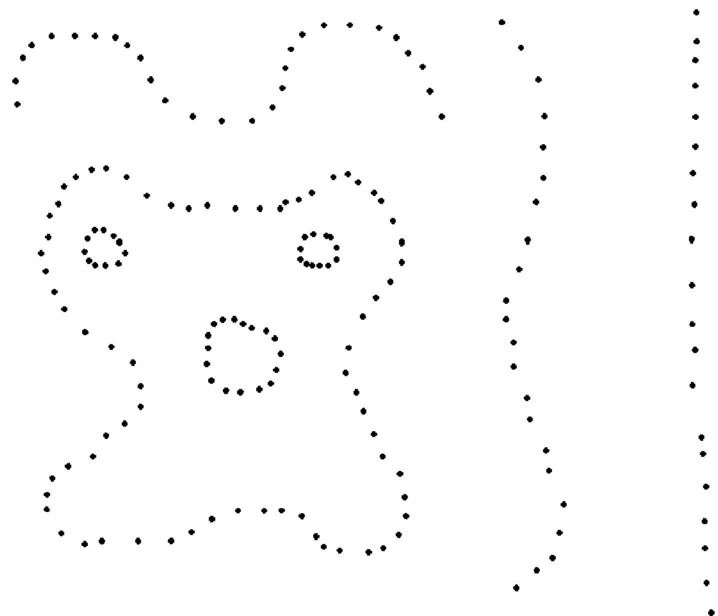


(a) sample Points

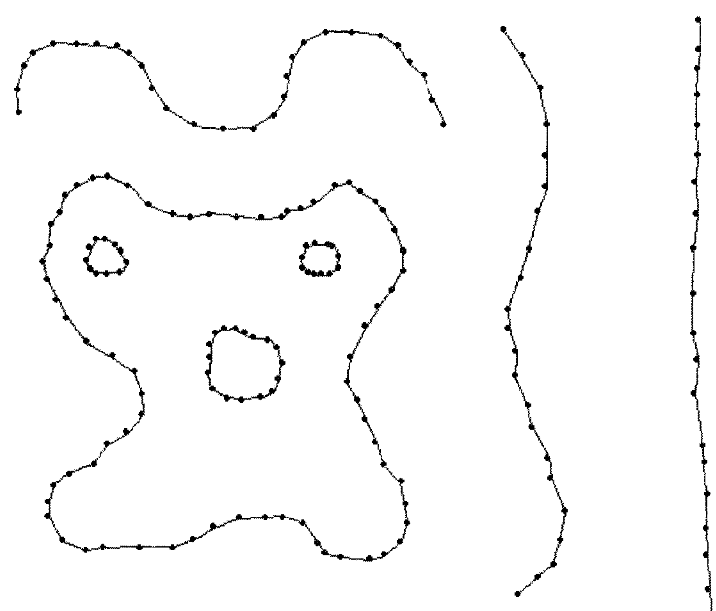


(b) Reconstructed Curve

Figure 3.16: *Example1- Sample point set and reconstructed curve*

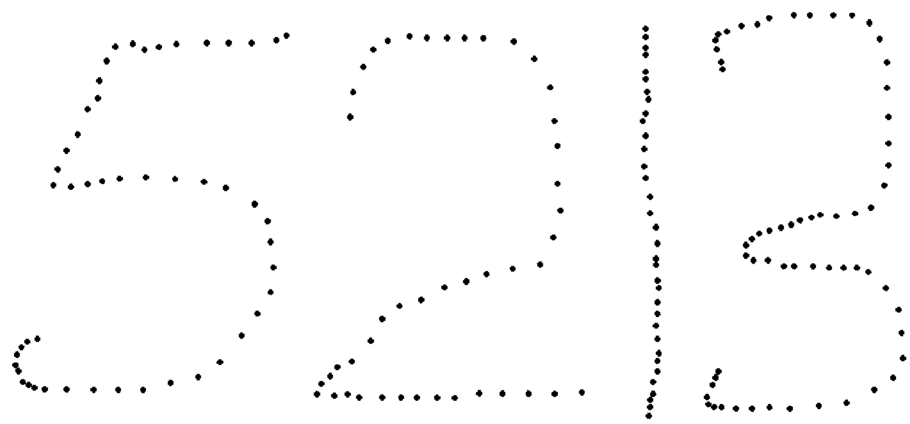


(a) sample Points

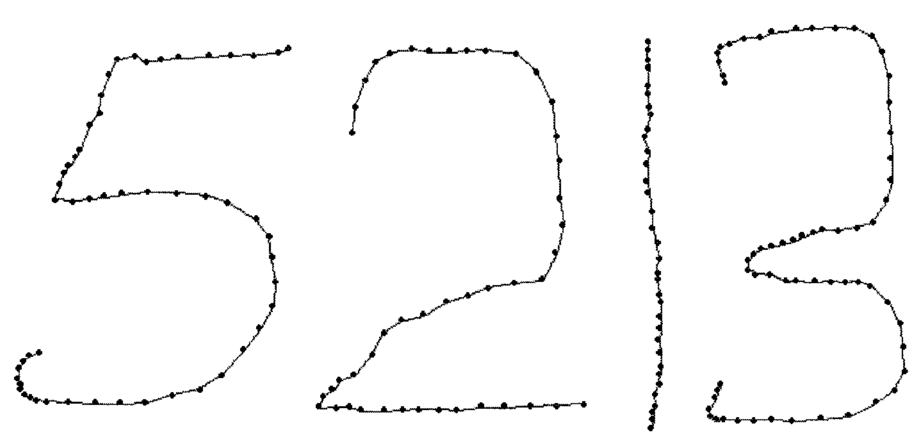


(b) Reconstructed Curve

Figure 3.17: *Example2- Sample point set and reconstructed curve*



(a) sample Points



(b) Reconstructed Curve

Figure 3.18: *Example3- Sample point set and reconstructed curve*

Chapter 4

Curve Reconstruction in Presence of Noise

In the previous chapter we discussed the problem of surface reconstruction when we have a noise-free sample from the original curve which we wish to reconstruct. Most guarantees on faithful reconstruction are dependent on the sampling conditions being satisfied. However in many practical situations the sample we receive is not noise-free. In this section we review some previous work that deal with noisy samples and then present a simple solution for the problem.

4.1 Introduction

Noise creeps into a sample point set during the process of collecting the sample. A sample point set that is input to a reconstruction algorithm is often obtained by a scanning process. It is during the scanning of the original curve that noise creeps in. These can originate from human error or machine error.

4.1.1 Data collection

An important application of the two problems of curve and surface reconstruction is reverse engineering. It is called reverse engineering because we start with an existing object, collect input points from it and then recreate a model. This exercise is often needed in industry to analyze the existing models and to study them for future modifications and improvements.

A sample input for the curve or 3D surface reconstruction problem is obtained by the use of fast and accurate methods like optical or laser ranger scanners, acoustic sensors and magnetic sensors. Data collection methods can be broadly classified into contact and non-contact types(see Fig. 4.1).

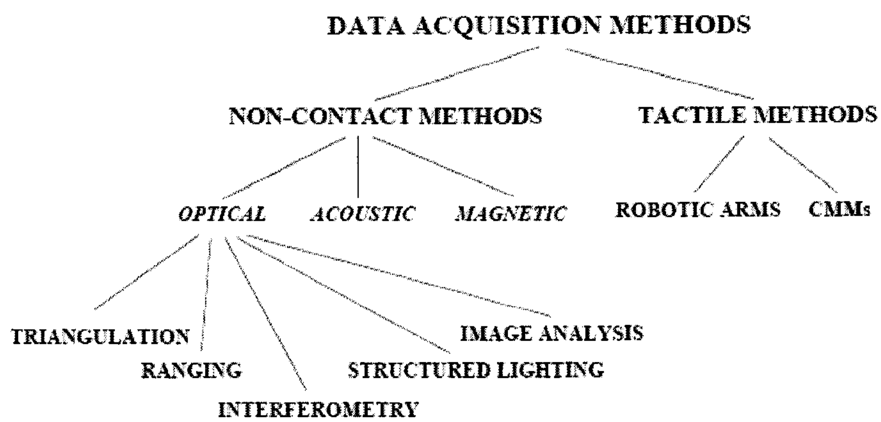


Figure 4.1: *Data Acquisition methods [37]*

There are many obstacles in the data collection process like -

- Calibration.
- Accuracy
- Accessibility.
- Occlusion.
- Multiple views.
- Noisy and incomplete data.

For more details about reverse engineering and data acquisition methods see Varady et al.[37].

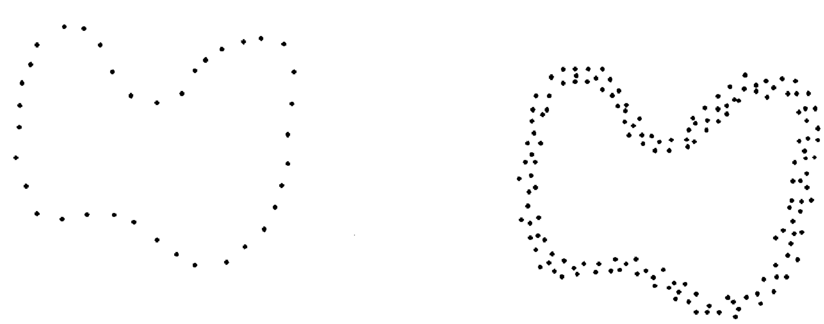
For the problem at hand we are interested in noise that creeps into the sample point set during the scanning procedure. Noise elimination for scanned data is a difficult problem and reconstructing a curve in the presence of noise in a sample point set is a challenging problem.

4.1.2 Noise Types

In general, noise in an input sample point set can be classified into two types according to where the noise is present with respect to the location of the original curve :

- noise points that lie close to the curve.
- noise points that lie far from the curve (outliers).

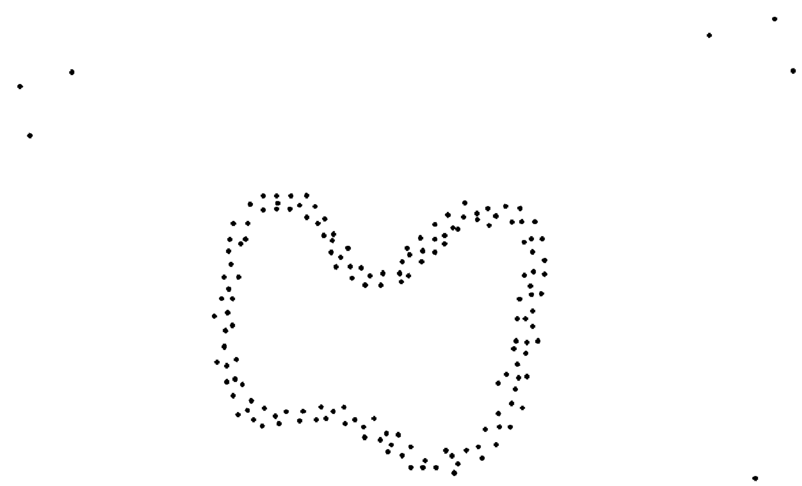
Most current algorithms address the first type of noise, that is noise that is present in close proximity to the original curve and assume that the input sample does not contain outliers.



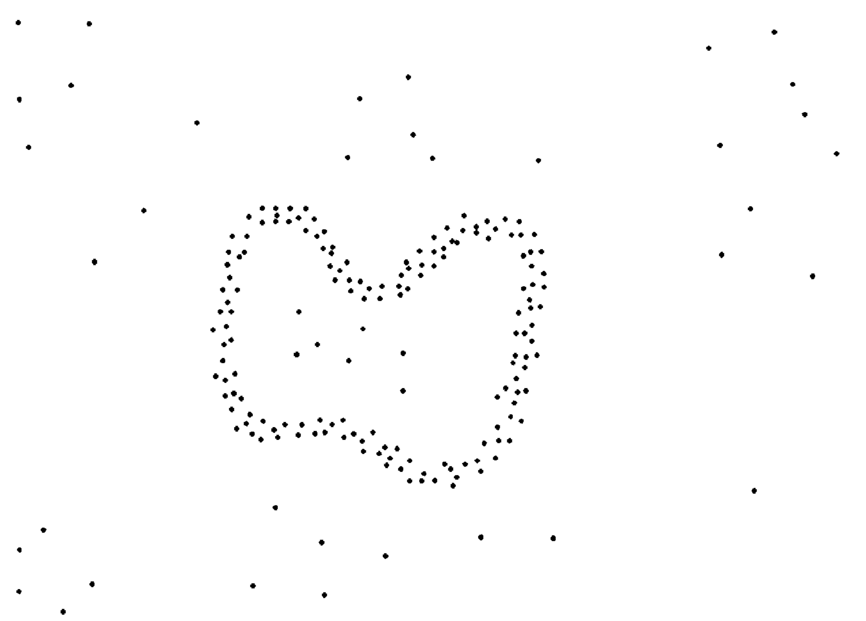
(a) A normal sample without noise

(b) A noisy sample with noise points close to the curve

Figure 4.2: *A normal sample and a noisy sample*



(a) A noisy sample with noise points close to the curve and some outliers



(b) Completely noisy sample with noise points interior to the curve, outliers and points close to the boundary

Figure 4.3: *Types of noisy samples*

4.2 Literature review

Investigating a solution to the curve reconstruction problem in presence of noise adds a new dimension to the curve reconstruction problem. Most of the proposed algorithms proceed by assuming a reasonable noise model and the input sample set is assumed to be consistent with this noise model.

Cheng et al.[10] propose an algorithm to reconstruct a collection of disjoint smooth curves from noisy samples. Their noise model is based on the following assumption:

- Absence of outliers in the sample point set.

  They enforce this by taking a normal sample and then introducing the first type of noise into the sample by perturbing the sample points uniformly in normal directions.

The algorithm works with the noisy sample created as above in three steps:

STEP1 (Point Estimation) In this step they determine a new point set that is less noisy than the original sample point set. For each sample point s in the sample set S, they determine a thin rectangle called $refined$(s) and the center of this rectangle they use as the new desired point $s*$. They do this by first finding a horizontal strip inside a big circle, $coarse(s)$ called coarse neighbourhood of s. The coarse neighbourhood is obtained by taking a point s as initial point and growing a circle around until the $coarse(s)$ is big enough to capture the noise strip as per the noise model. They quantify the ratio of radius of $coarse(s)$ and $width(strip(s))$ to be greater than or equal to a constant ρ. Then they find another strip which is perpendicular to the current strip to form the rectangle which they rotate and adjust to find the smallest bounding rectangle for the points captured inside the rectangle. The center of this rectangle is the new point $s*$ desired. This point $s*$ replaces the point s on the pruned set of points. When this step is done they reduce the scatter of the noise points on both sides of the curve therefore reducing the noise. However a reconstruction at this stage will produce a highly jagged polygonal curve.

STEP2 (Pruning) This step is used as a remedy to the problem of reconstructing the curve from all points $s*$ obtained above. In this step they sort the points $s*$ in decreasing order of width ($refined(s)$). Then the sorted list is scanned and a subset of center points $s*$ is selected. While selecting a center point $s*$ they delete all center points $u*$ from the sorted list such that $\| s^* - u^* \| \leq width(refined(s)))^{1/3}$.

STEP3 (Reconstruction) Finally they use NN-Crust algorithm on the selected points from step 2 to perform the reconstruction.

Most algorithms proposed in the literature dealing with noisy input sample are for surface reconstruction. However in many cases this problem is first discussed in 2D and then generalized to three dimensions. For example, Dey and Goswami [16] propose an algorithm for surface reconstruction in the presence of noise. Fig. 4.4 illustrates their approach.

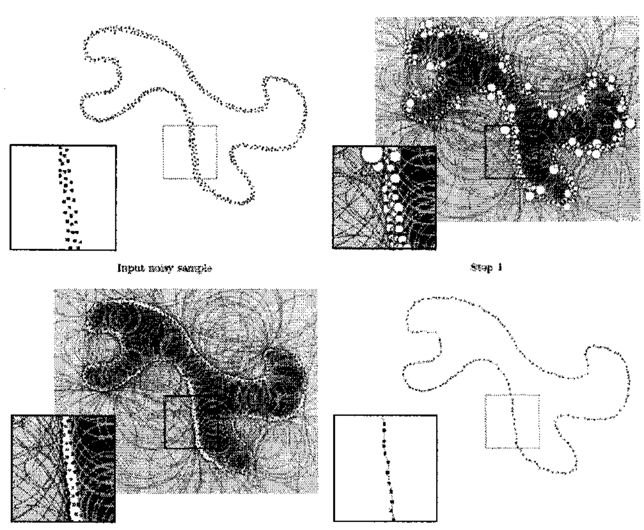


Figure 4.4: *Step1 : Big Delaunay Balls(shaded) are separated from small ones(unshaded), Step 2: Outer and Inner big Delaunay balls are separated, Step 3: Only points on the outer balls are retained and the curve(surface) is reconstructed from them [16]*

They provide some theoretical guarantees in a noise model which allows the input points to be scattered around the sampled surface and the range of the scatter is constrained by the local feature size. The algorithm makes use of Voronoi diagram and Delaunay triangulations of the input points and uses the power crust algorithm [4]. For some other algorithms on surface reconstruction from noisy samples see [30],[33],[38].

4.2.1 Noise Model

For our experiments the noise model is built from an ϵ- sample obtained from an unknown curve. The noise points are generated from the reference ϵ-sample and are located around the generating sample point. Since an ϵ-sample imposes the condition that at any point p on the curve there must be a sample point at a maximum distance of $\epsilon * f(p)$, we impose a restriction that the noise points are located on a strip around the original curve within a maximum distance $< 1/2 * \epsilon * f(p)$ from the curve. We estimate $f(p)$ by computing the minimum distance to the Voronoi vertices of the Voronoi Polygon for the generating sample point. Within the strip around the curve, the noise points are placed at random distances from the generating sample point.

The noise model that we use imposes the following constraints on the input sample:

- The noise points lie close to the boundary of the original curve and no outliers are present.
- At any point p on the original curve the noise points are located at a maximum distance $d < 1/2 * \epsilon * f(p)$ from the point p, with $\epsilon < 1/5$.

Further we estimate a constant ρ for the noise model. This constant ρ is calculated as follows:

- For each sample point we calculate the minimum distance from the sample point to the Voronoi vertices of the Voronoi polygon of that sample point.
- Compute the average of these minimum distances over all sample points and assign it to ρ.

4.3 An algorithm for noise filtering

When we obtain a noisy sample from a source, the reconstruction is difficult as we have no idea as to which are the points from the original curve and which are the noise points that crept in during the scanning process. In such a case we try to filter the input sample and obtain a set of points that would be useful in approximating the original curve. In this section we propose a simple heuristic to filter a noisy sample. This algorithm is called the curveReconstructionwihNoise algorithm or $CRWN$ for short.

We start with a noisy set of points as in Fig. 4.5(only a part of curve is shown).

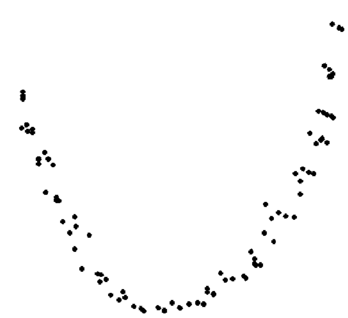


Figure 4.5: *Noisy point set showing part of a curve*

The first step of the algorithm is to find a point p from the input point set S, that can be used as a starting point in the filtering process. The point with the minimum y co-ordinate is chosen as the starting point and is called the seed point SP. This is done to keep the choice of seed point SP as simple as possible. Then we draw a circle around SP. This circle is called the PRUNECIRCLE C. The radius r of the PRUNECIRCLE is less than $f(p)$ /2. Fig. 4.6 shows the PRUNECIRCLE and the seed point. Fig. 4.7 shows the PRUNECIRCLE enlarged. For our experiments, we generate the noisy sample and the size of the PRUNECIRCLE is dependent on the noise model. we take $\rho/2$ as the radius of the PRUNECIRCLE as it performs well over a variety of sample point sets.

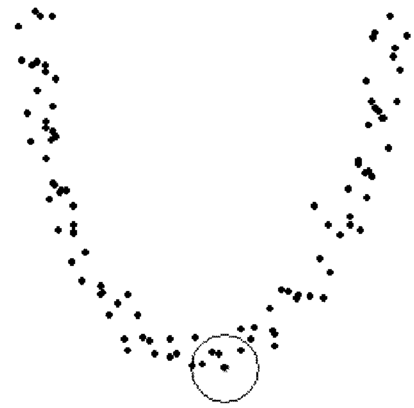


Figure 4.6: *PRUNECIRCLE and SEED POINT on the Noisy point set*

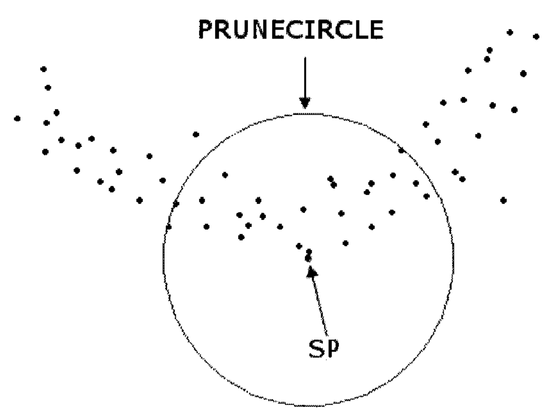


Figure 4.7: *ENLARGED view of PRUNECIRCLE and SEED POINT on the Noisy point set*

The goal is to select a single point from the interior of C as our filtered point and reject all other points points in the interior of C. Fig. 4.8 shows the points inside C. To chose the filtered point $p*$, we find a pair of points inside the current PRUNECIRCLE which are located farthest from each other in the interior of C. The mid point of the line joining these extreme points inside the PRUNECIRCLE C is taken as the new filtered point. Fig. 4.9 illustrates this development.

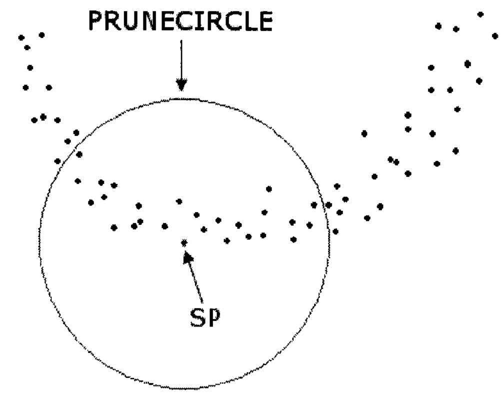


Figure 4.8: *Blue dots represent points inside PRUNECIRCLE*

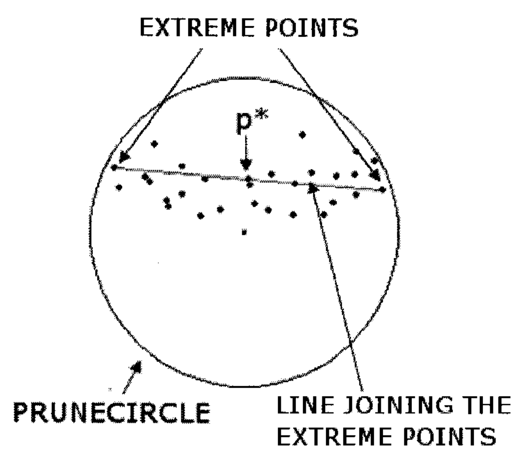


Figure 4.9: *Extreme points and Mid point of the line joining them*

This new filtered point $p*$ is added to the filtered point set S' if it passes the *validity test.* All other points inside the current PRUNECIRCLE are discarded. The *validity test* imposes a constraint on the filtered point $p*$ which can be stated as follows:

- The new filtered point $p*$ must not lie inside the previous PRUNECIRCLE. Since the current and previous PRUNECIRCLE overlap each other, its necessary that the new filtered point $p*$ does not belong to the intersection of the current PRUNECIRCLE and the previous PRUNECIRCLE.

If the new filtered point $p*$ lies inside the previous PRUNECIRCLE we reject it as it fails the *validity test* and consider the next highest distance between a pair of points inside the current PRUNECIRCLE. This process is repeated until we have a new filtered point $p*$ that passes the *validity test*. When a filtered point $p*$ passes the *validity test*, it is added to the filtered point set S'. Fig. 4.10 illustrates the case where a filtered point can lie within the previous PRUNECIRCLE. The mid-point q of the line joining the extreme points a and b is rejected as it fails the validity test and consider the next highest distance between a pair of points a and c. The mid-point of the line joining inside the current PRUNECIRCLE.

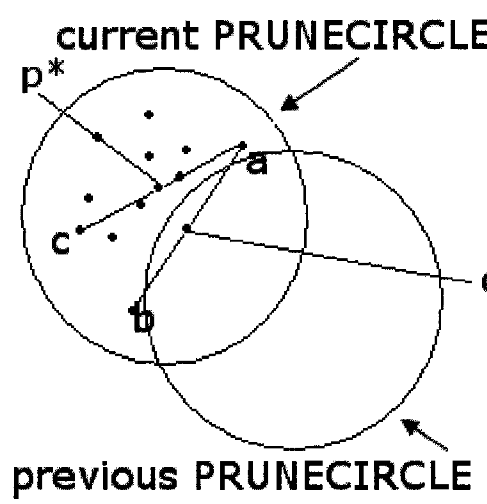


Figure 4.10: *Validity test on the new filtered point: q is rejected as it falls inside the previous PRUNECIRCLE and the mid-point of ac is takes as p*.*

The next step is to update the seed point SP to a point exterior to C. Among the remaining points of the noisy point set S the point closest to the current PRUNECIRCLE is the new seed point. Points inside the current PRUNECIRCLE are not considered again as they are discarded. Fig. 4.11 shows the new seed point SP in green and its the closet point to the current PRUNECIRCLE C shown in red. It also shows the pruned point $p*$ in blue. All other points inside the current circle have been removed at this point.

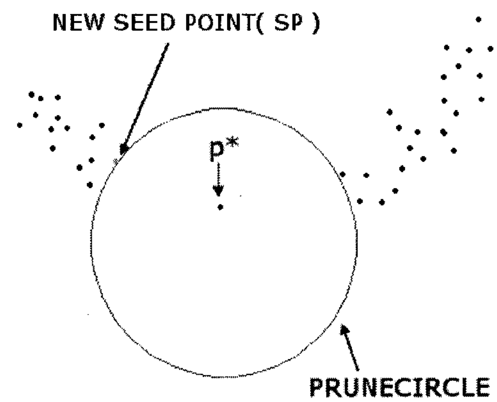


Figure 4.11: *Filtered point shown in blue inside circle and the new seed point shown in Green is closest to the circle*

Fig. 4.12 shows how this process occurs on the noisy point set (part of an unknown curve).

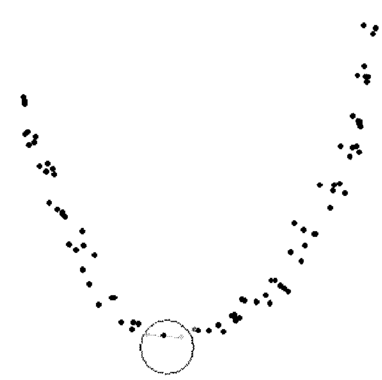


Figure 4.12: *The entire process on a Noisy point set (Only part of the unknown curve is shown)*

The formal description of the CRWN algorithm is given below.

Algorithm $CRWN$

Input: A set of sample points with noise S from an unknown smooth curve C.

Output: A polygonal reconstruction of C.

Step 1. Find the point p of S which has the minimum value of y co-ordinate(lowest point) and use it as a seed point SP.

Step 2. Draw a PRUNECIRLCE of radius r around the seed point, where $r < f(p)\ /2$.

Step 3. Find a pair of points inside the PRUNECIRCLE at a maximum distance from each other.

Step 4. Compute the mid point of the line joining the extreme points and perform the *validity test* to find the new filtered point $p*$.

Step 5. Add the new filtered point $p*$ to S' and discard all other points inside the circle.

Step 6. Set the new seed point SP to a sample point which lies outside the current PRUNECIRCLE and is closest it.

Step 7. Repeat steps 2 to 6 over the remaining points of the noisy sample point set.

Step 8. Use CR algorithm on the filtered point set S' to find the polygonal reconstruction of the unknown curve.

4.4 Experiments with noisy samples

The algorithm was implemented in Java and experiments were performed on a variety of sample points. Fig. 4.13(a) shows a normal sample point set and Fig. 4.13(b) shows a noisy sample point set drawn from the same unknown curve. Fig. 4.13(c) shows the pruned point set S' obtained after applying the $CRWN$ algorithm to the noisy point set S.

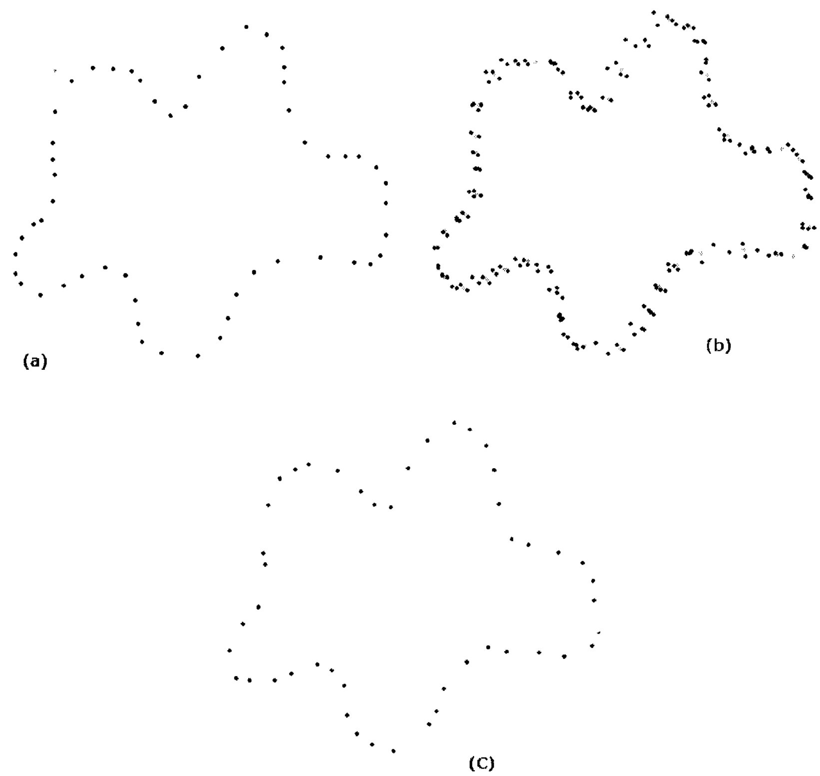


Figure 4.13: *Example 1: A Normal sample from an unknown Curve (a), A noisy Sample from the same curve (b), filtered Points obtained by filtering the Noisy Sample (c)*

The following figures show the results of application of the noise filtering algorithm on different sets of sample noisy point sets drawn from various unknown curves and collection of curves. Each figure shows the noisy point set S and the pruned point set S' after application of the pruning algorithm $CRWN$ described above.

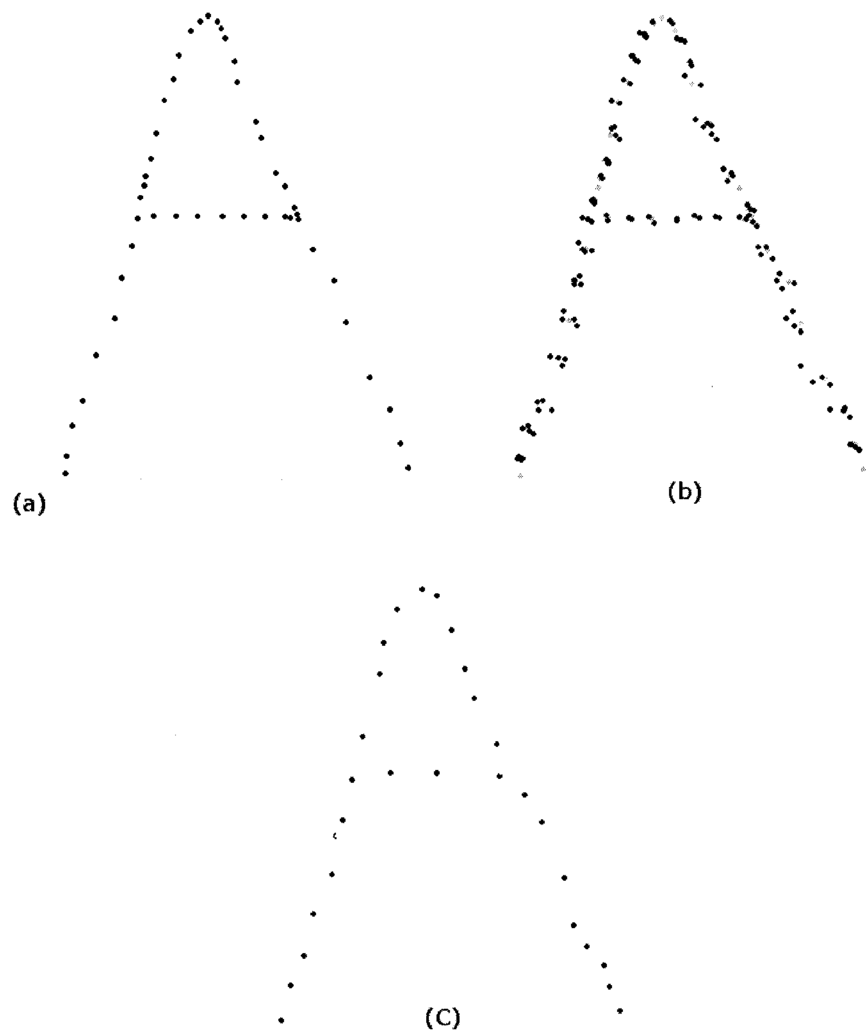


Figure 4.14: *Example 2: A Normal sample (a), noisy Sample (b), and Points obtained after filtering (c)*

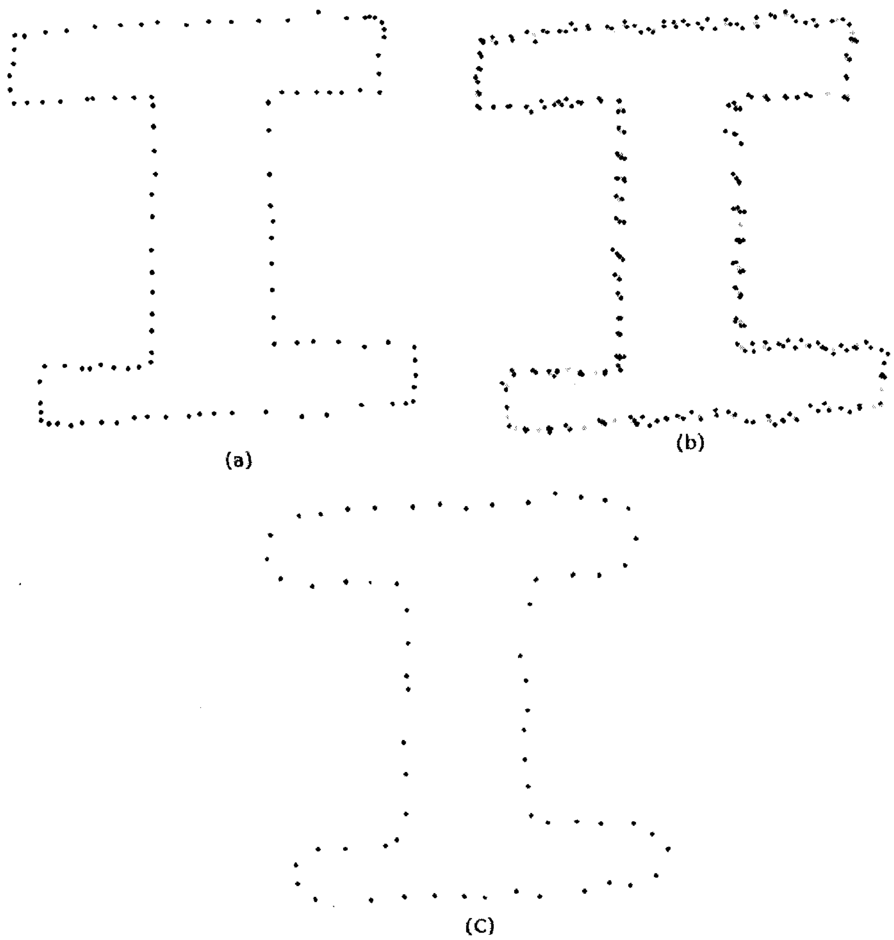


Figure 4.15: *Example 3: A Normal sample (a), noisy Sample (b), and Points obtained after filtering (c)*

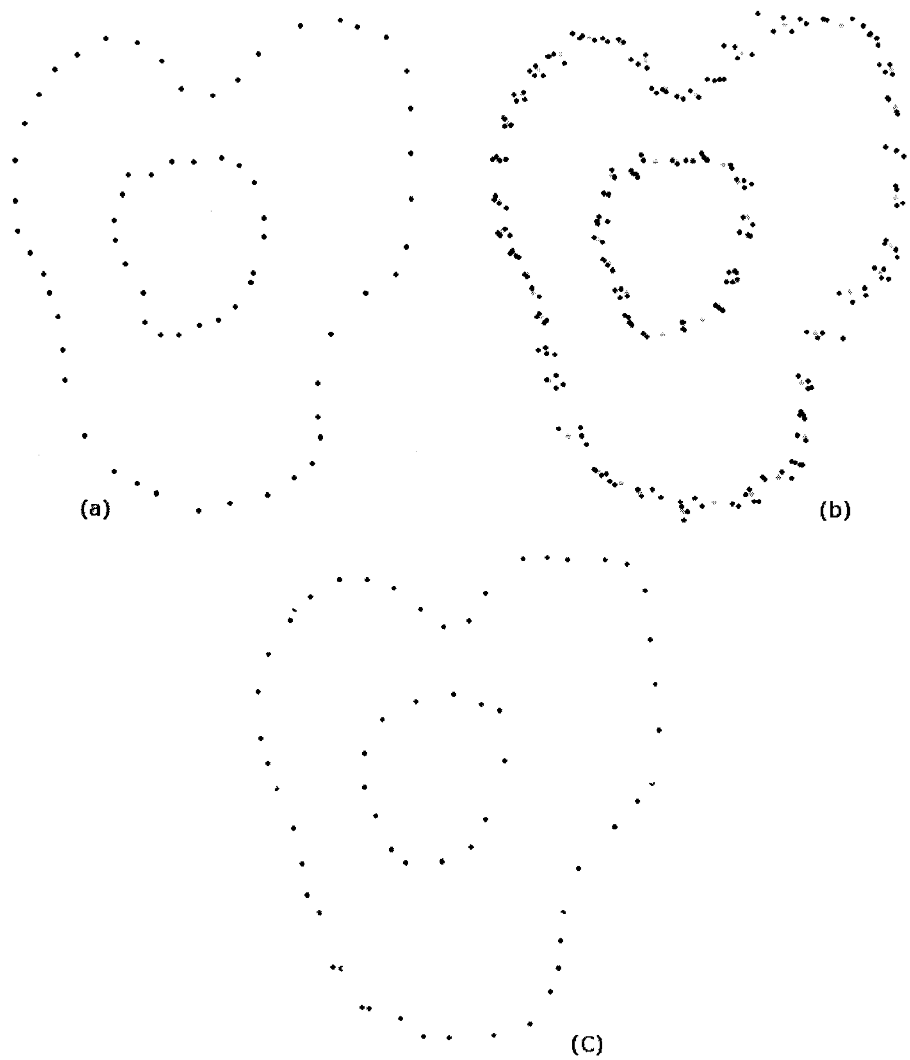


Figure 4.16: *Example 4: A Normal sample (a), noisy Sample (b), and Points obtained after filtering (c)*

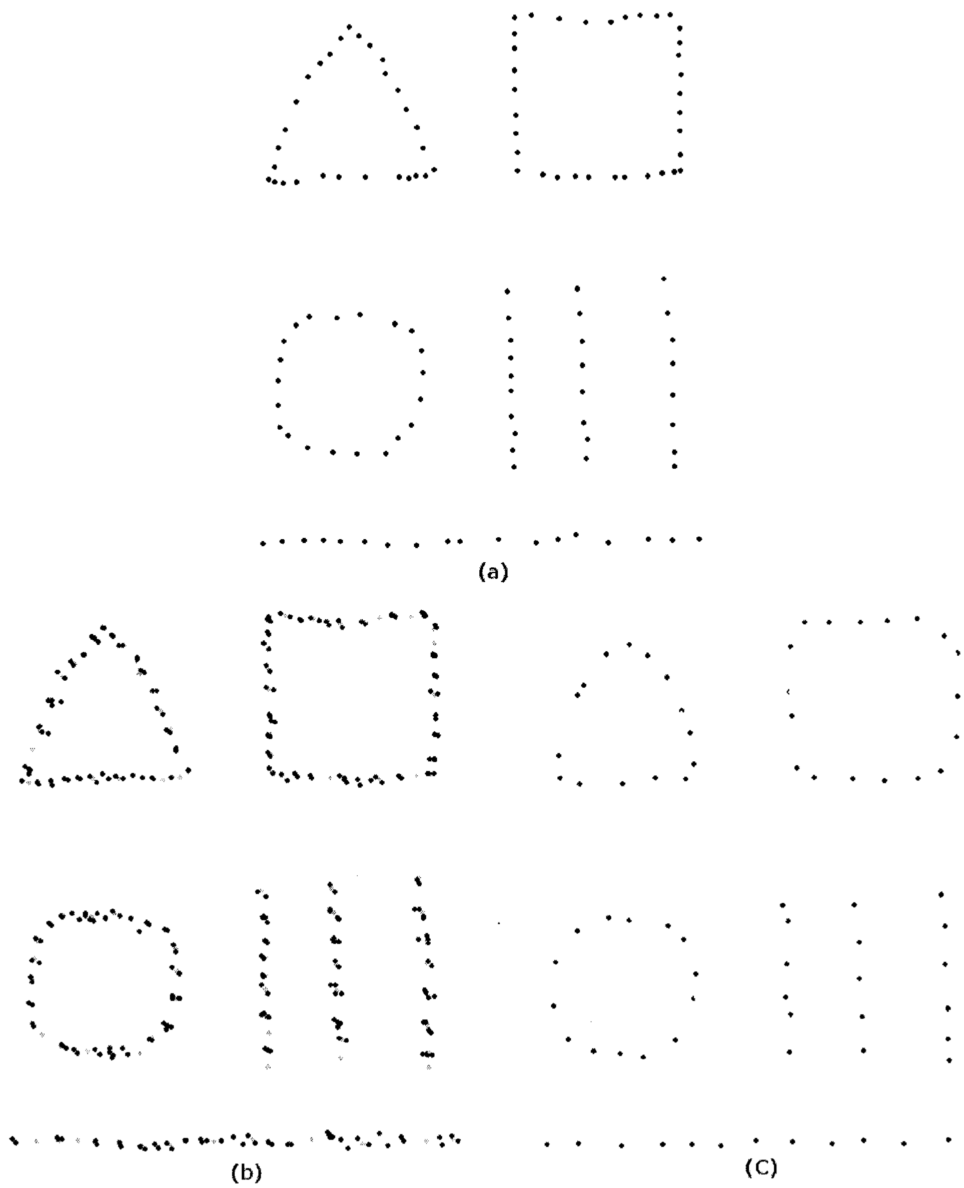


Figure 4.17: *Example 5: A Normal sample (a), noisy Sample (b), and Points obtained after filtering (c)*

4.5 Implementation Details

The Algorithm was implemented with Java. The following pages show screen-shots from the implementation applet. The interface of the applet is divided into two halves by a vertical line. The left hand side is used as a reference to build the noise model. The noise points are induced to the right hand side and the reconstruction algorithm ($CRWN$) described above is used to filter the points and find a reconstruction which is similar to the reference curve.

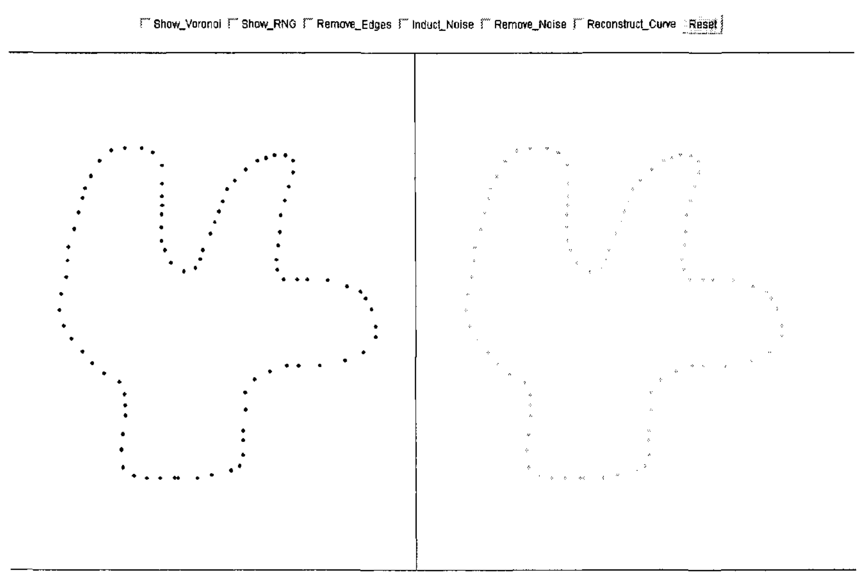


Figure 4.18: *Sample points entered by the user. The left hand side is used as the reference. The sample points are redrawn on the right hand side.*

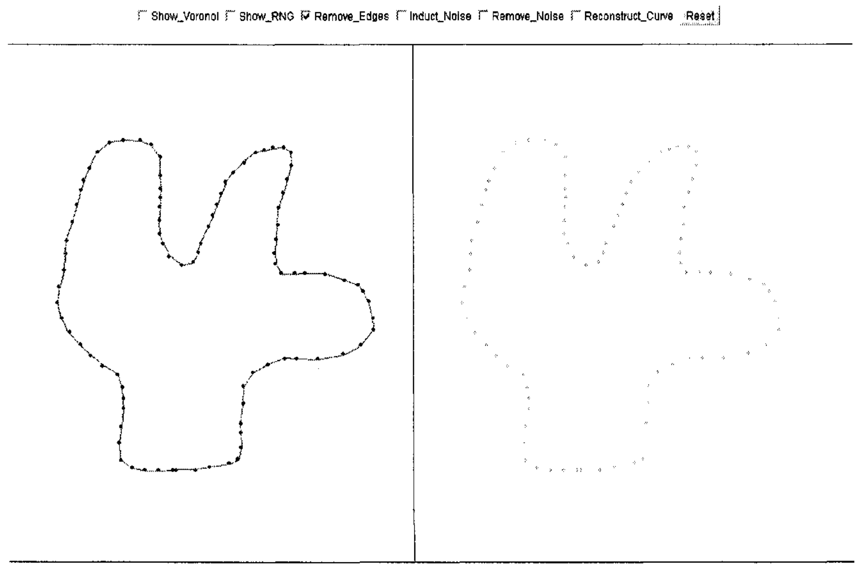


Figure 4.19: *The unknown curve is reconstructed from the sample points entered and it serves as a reference to build the noise model.*

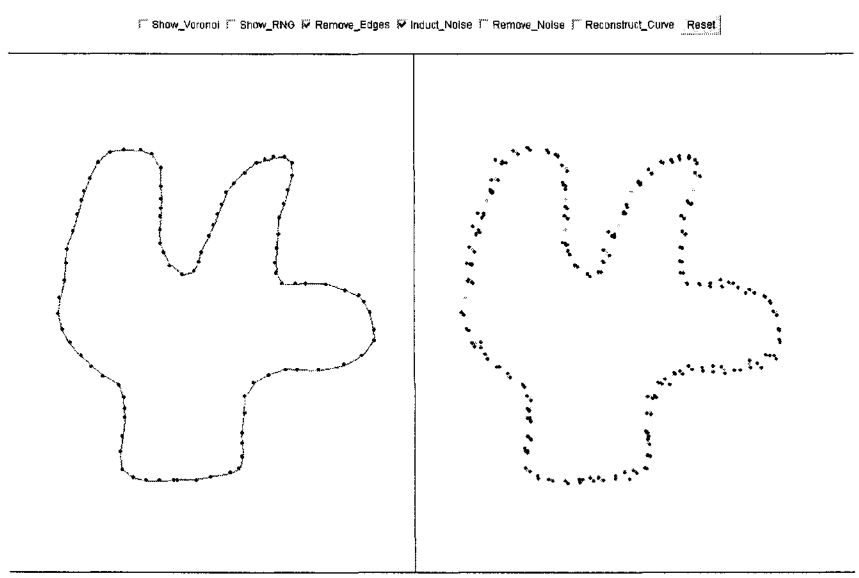


Figure 4.20: *Noise points generated from the reference sample points on the left hand side are induced on the right hand side.*

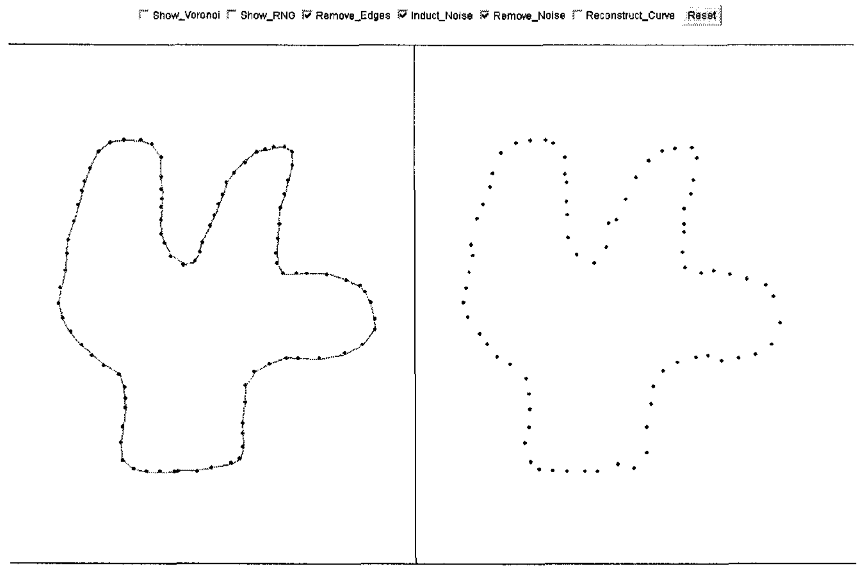


Figure 4.21: *Filtered points on the right hand side after the filtering algorithm is applied.*

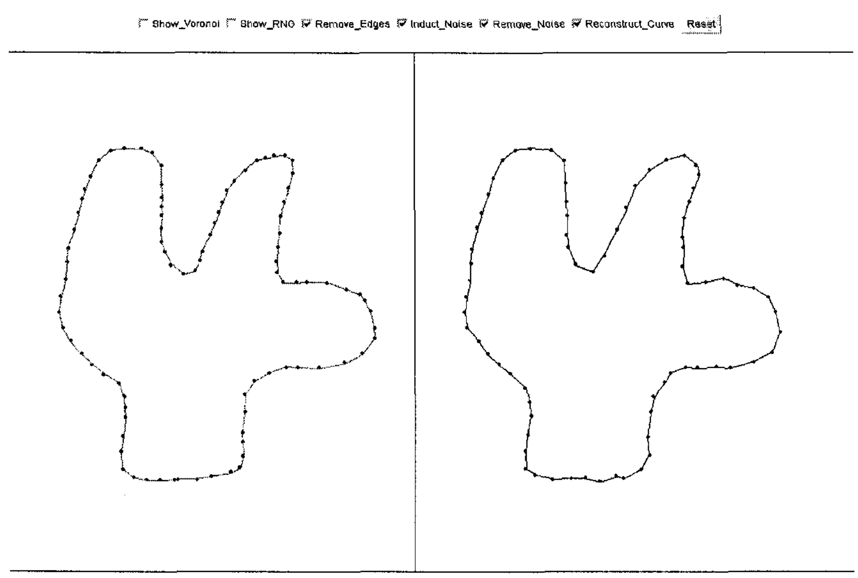


Figure 4.22: *The unknown curve is reconstructed from the filtered points(right hand side). The reference curve is shown on the left hand side.*

Chapter 5

Conclusion

Most algorithms on curve reconstruction concentrate on finding a reconstruction from a pure sample without any noise points. The algorithms cater to a variety of curves on which they can be applied and they impose restrictions on the quality of the sample which is essential for a faithful reconstruction. Further most attempts at solving the curve reconstruction problem are Delaunay based approaches and involve making use of a subgraph of the Delaunay triangulation as a starting point in the reconstruction process.

We have proposed a novel curve reconstruction algorithm that is extremely simple to understand and implement while being very effective for reconstruction. A different subgraph of the Delaunay triangulation known as Relative Neighbourhood Graph is used here the starting point. The advantage is that it has fewer redundant edges than the Gabriel graph. The time complexity is in $n \log(n)$ time, where n is the number of sample points. The results of experiments compare very favorably to examples in the literature. This work has been reported at ISVD'06 [31].

We have also explored the non-ideal situation where the sample obtained is not pure, that is it has noise or outliers in it. A simple heuristic has been proposed to eliminate noise points from such a faulty sample. This works by taking a seed point, which is the point with lowest y axis coordinate and use a moving circle to go through the entire point set to filter noise points. Once we have a filtered set of points we can apply the normal curve reconstruction algorithm suggested above to find a faithful reconstruction.

Bibliography

[1] E. Althaus and K. Melhorn. TSP-based curve reconstruction in polynomial time. In *Proc. 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 686–695, 2000.

[2] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. In *Graphical Models and Image processing*, volume 60, pages 125–135, 1998.

[3] N. Amenta, M. Bern, and M. Kamvysselis. A New Voronoi-Based Surface Reconstruction Algorithm. In *Proc. 25th Annu. Conf. on computer graphics and interactive techniques*, pages 415–421, 1998.

[4] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geom. Theory and Appl.* , 19(2-3), pages 127–153, 2001.

[5] D. Attali. r-regular shape reconstruction from unorganized points. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 248–253, 1997.

[6] F. Aurenhammer. Voronoi diagrams a survey of a fundamental geometric data structure. In *ACM Computing Surveys*, 23(3), pages 345–405, 1991.

[7] F. Bernadini and C. Bajaj. Sampling and Reconstructing Manifolds Using Alpha-Shapes. *Proc. 9th Canadian Conf.Comput. Geom.*, pages 193–198, 1997.

[8] F. Bernadini, C. Bajaj, J. Chen, D. R.Schikore. Automatic Reconstruction of 3D CAD Models from Digital Scans. *International Journal of Comput. Geom. and Appl.*, Pages 327–370, 1999.

[9] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–338, 1992.

[10] S. W. Cheng, S. Funke, M. Golin, P. Kumar, S. H. Poon and E. Ramos. Curve Reconstruction from noisy samples. *Computational Geom. Theory and Appl.*, 2004.

[11] D. Coppersmith and M. Sudan. Reconstructing curves in three (and higher) dimensional space from noisy data. *35th Annu. ACM Sympos. on Theory of Computing*, pages 136–142, 2003.

[12] T. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *ACM-SIAM Symposium Discr. Algorithms*, pages 893–894, 1999.

[13] T. Dey, K. Mehlhorn, and E. A. Ramos. Curve reconstruction : connecting dots with good reason. In *Proc. 15th Annual ACM Sympos. Comput. Geom.*, pages 197–206, 1999.

[14] T. K. Dey and R. Wenger. Reconstructing curves with sharp corners. *Computational Geom. Theory and Application*, 19(2-3), pages 89–99, 2001.

[15] T. K. Dey. Curve and Surface Reconstruction. *Chapter - Handbook of disc. and computational Geom., Goodman and O' Rourke, CRC press, 2nd edition*, 2004.

[16] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 330–339, 2004.

[17] H. Edelsbrunner, D. G. Kirkpatrick and R. Seidel. On the shape of a set of points in the plane. *IEEE transactions on information theory*, 29(4), pages 551–559, 1983.

[18] L. H.de Figueiredo and J. de Miranda Gomes. Computational morphology of curves. In *Visual Computer.*, 11:105–112, 1995.

[19] S. Funke and E. A. Ramos. Reconstructing a collection of curves with corners and endpoints *Proc. 12th ACM-SIAM Stmpos. on Discrete Algorithms*, pages 344–353, 2001.

[20] J. Giesen. Curve reconstruction, the travelling salesman problem and Menger's theorem. *Discr. Comput. Geom.*, 24:577–603, 2000.

[21] J. Giesen. Curve reconstruction. *Ph.D. Thesis, Swiss Federal Inst. of Technology*, 2000.

[22] C. Gold. Crust and Anti-Crust: A One-Step Boundary and Skeleton Extraction Algorithm. *Proc. 15th Annual ACM Sympos. Comput. Geom.*, pages 189–196, 1999.

[23] S. Guha and S. D. Tran. Reconstructing curves without delaunay computation. *Algorithmica*, 42:75–94, 2005.

[24] H. Hiyoshi. Closed Curve Reconstruction from Unorganized Sample Points *3rd Int. Sympos. on Voronoi Diagrams* , pages 122–131, 2006.

[25] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80(9):1502–1517, 1992.

[26] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *systematic Zoology*, pages 259–278, 1969.

[27] I. K. Lee. Curve reconstruction from Unorganized Points. *Computer aided Geometric design*, 17:161–167, 2000.

[28] T. Lenz. Simple reconstruction of non-simple curves. *Tech. Report B 05-02, Freie Universitat Berlin*, 2005.

[29] T. Lenz. Reconstructing collections of arbitrary curves. *Proc. 21st Annual ACM Sympos. Comput. Geom.*, pages 366–367, 2005.

[30] B. Mederos, N. Amenta, L. Velho and L. H. de Figueiredo. Surface reconstruction from noisy point clouds. *Eurographics Symposium on Geometry processing*, 2005.

[31] A. Mukhopadhyay and A. Das. An RNG-Based Heuristic for Curve Reconstruction. *3rd Int. Sympos. on Voronoi Diagrams* , pages 246–251, 2006.

[32] J. O'Rourke, H. Booth and R. Washington. Connect-the-Dots: a new heuristic. *Computer Vision, Graphics and Image Processing*, 39:258–266, 1984.

[33] S. H. Poon. Curve and surface reconstruction from noisy samples. *Ph.D. Thesis, Hong Kong University of Science and Technology*, 2004.

[34] C. K.Tang and G. Medioni. Inference of integrated surface, curve and junction descriptions from sparse 3D data. *IEEE Trans. on Pattern analysis and Machine intelligence*, 20(11), Pages 1206–1223, 1998.

[35] G. Taubin and R. Ronfard. Implicit Simplicial Models for Adaptive Curve Reconstruction. *IEEE Trans. on Pattern analysis and Machine intelligence*, 18(3), Pages 321–325, 1996.

[36] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.

[37] T. Varady, R. R. Martin and J. Cox. Reverse engineering of geometric models–an introduction. *Computer-Aided Design*, 1997.

[38] H. Xie, K. McDonnell, H. Qin. Surface reconstruction of noisy and defective data sets. *IEEE Visualization*, pages 259–266, 2004.

Vita Author

Name	:	Augustus Kumar Das.
Date of Birth	:	24^{th} August, 1978.
Place of Birth	:	Cuttack, India.
Education	:	

- Master of Science, Computer Science, 2007. University of Windsor, Canada.
- Bachelor of Engineering, Computer Science, 2001. Utkal University, India.